

ND-A193 043

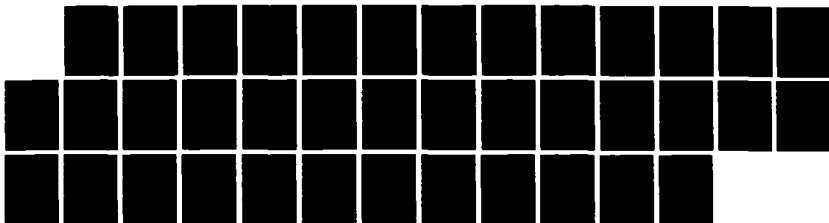
A USER'S/PROGRAMMER'S MANUAL FOR TANKET(0) NAVY
RESEARCH LAB WASHINGTON DC T F SNEAR 06 MAY 88
NRL-NR-6201

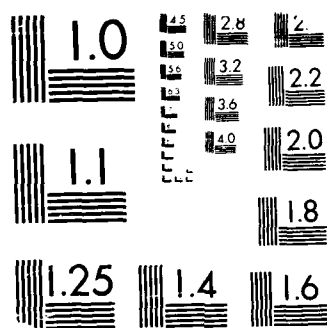
1/1

UNCLASSIFIED

F/G 12/5

NL





MICROCOPY RESOLUTION TEST CHART
 NATIONAL BUREAU OF STANDARDS-1963-A

Naval Research Laboratory

Washington, DC 20375-5000

DTIC FILE COPY



NRL Memorandum Report 6201

A USERS'/PROGRAMMERS' MANUAL FOR TWAKE

THOMAS F. SWEAN, JR

*Center for Hydrodynamic Developments
Laboratory for Computational Physics and Fluid Dynamics*

May 6, 1988

AD-A193 043

DTIC
ELECTE
JUN 06 1988
S **D**
H

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NRL Memorandum Report 6201			5. MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION Naval Research Laboratory		6b. OFFICE SYMBOL (If applicable)		7a. NAME OF MONITORING ORGANIZATION	
6c. ADDRESS (City, State, and ZIP Code) Washington, DC 20375-5000				7b. ADDRESS (City, State, and ZIP Code)	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Office of Naval Research		8b. OFFICE SYMBOL (If applicable)		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State, and ZIP Code) Washington, DC 20375		10. SOURCE OF FUNDING NUMBERS			
		PROGRAM ELEMENT NO. 62111N		PROJECT NO.	TASK NO. RU11W 71-801
				WORK UNIT ACCESSION NO. DN156-264	
11. TITLE (Include Security Classification) A Users/Programmers' Manual for TWAKE					
12. PERSONAL AUTHOR(S) Swean, T.F., Jr.					
13a. TYPE OF REPORT Interim		13b. TIME COVERED FROM TO		14. DATE OF REPORT (Year, Month, Day) 1988 May 6	
15. PAGE COUNT 38					
16. SUPPLEMENTARY NOTATION <i>1 page</i>					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP	Programmers' manual CMC-3DPNS, Turbulent ship wake		
			Finite-element method TWAKE		
19. ABSTRACT (Continue on reverse if necessary and identify by block number)					
<p>→ A detailed description is given of the procedures required to initialize the computer code TWAKE for a class of surface ship wake problems. The code is a modified combination of variants of the CMC-3DPNS finite-element solver and is applicable to the steady, Reynolds-averaged, parabolic Navier-Stokes equations with a $k-\epsilon$ turbulence closure model. General descriptions are given of the computational model and the logical path through the execution modules. Samples are given of data resulting from all output units. The program has been executed for the given data on Cray, Hewlett-Packard, and Digital Equipment computer systems with identical results to 4 significant figures for single precision computations.</p> <p><i>turbulent ship wake, finite element</i></p>					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL Thomas F. Swean, Jr.			22b. TELEPHONE (Include Area Code) 202-767-2114		22c. OFFICE SYMBOL Code 4430

CONTENTS

1. INTRODUCTION	1
2. OVERVIEW OF THE COMPUTATIONAL MODEL	1
3. DESCRIPTION OF COMMAND DATA INPUT	4
4. OVERVIEW OF THE EXECUTION MODULE	11
5. CLOSING REMARKS	14
6. ACKNOWLEDGEMENTS	14
7. REFERENCES	14
TABLES	15
APPENDICES	17

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

A USERS'/PROGRAMMERS' MANUAL FOR TWAKE

1. INTRODUCTION

The computer program TWAKE is a modified version of the CMC-3DPNS program described in Refs. [1-4]. The parent code is available in the NASA/COSMIC software library (Ref. [8]) and, under separate contracts, variations of the original code have been acquired by the Naval Research Laboratory and the David Taylor Research Center. At NRL the program has been used to simulate a variety of turbulent, two and three-dimensional, self-propelled and drag wakes (Refs. [5-7]). In addition, several unpublished calculations of surface ship wakes and two-dimensional turbulent jets have been completed. Comparisons of the calculations with experimental data have consistently shown good quantitative agreement with respect to axial evolutions of characteristic (single-point) flow parameters such as maximum velocity deficit (excess) and maximum turbulence kinetic energy. In addition, the simulations of the experiments of Ref. [9] reproduced the qualitative behavior of major flow structures in the cross-plane.

In order to complete these numerical simulations, it has on occasion been necessary to alter and merge portions of the Navy-procured variants mentioned above. From these efforts has evolved a version of CMC-3DPNS which may be described as being "ship wake specific" while retaining the flexibility of the original program through the use of parameters defined during input.

NRL has been tasked to deliver the modified software to the OCNR (Code 12) Ship Wake Detection Program. This report is intended to serve as a programmers'/users' manual in support of that task. It should be considered as supplemental to Refs. [2-3] and this report will assume advance familiarity with that literature. The subsequent sections will describe the fundamentals of using TWAKE to simulate a class of self-propelled wake flows near a free surface. The physical approximations employed to derive the conservation equations as well as the mathematical concepts involved in their discretization are described in Refs. [1,4] and will not be repeated herein.

2. OVERVIEW OF THE COMPUTATIONAL MODEL

The conservation equations to be considered are the steady, three-dimensional, time-averaged (in the turbulence sense) parabolic Navier-Stokes equations. The effects of turbulence are described using modelled transport equations for the turbulence kinetic energy and the isotropic dissipation function along with an anisotropic closure for the turbulent stresses. Since the continuity equation is not parabolic, additional formulational steps are necessary to render the equation set amenable to a streamwise-marching numerical solution technique. As is usual the divergence of the transverse momentum equations provides an equation for the static pressure. The pressure field that satisfies this Poisson equation consists of complementary and particular parts with the complementary solution satisfying the homogeneous part of the original equation. An additional equation is provided by the definition of a harmonic function which is forced to zero (within a defined tolerance) as the continuity equation is satisfied. These equations and the ordering analysis leading to their derivation may be found in Ref. [4].

The computer program has been designed to consider the simultaneous convection and diffusion of up to 30 arbitrary scalar field variables, q_i . The general form of the equation system is

$$L(q_i) = u_1 \frac{\partial q_i}{\partial x_1} + c_i u_l \frac{\partial q_i}{\partial x_l} - d_i \frac{\partial}{\partial x_l} \left(\kappa_l \frac{\partial q_i}{\partial x_l} \right) + s_i = 0, \quad (1)$$

with boundary conditions

$$l(q_i) = a_{1il} q_i + a_{2il} \frac{\partial q_i}{\partial x_l} \tilde{n}_l + a_{3il} = 0, \quad (2)$$

and initial conditions

$$q_i(0, x_l) \equiv q_i^0(x_l). \quad (3)$$

In these equations x_1 is the primary flow (marching) direction, the subscript l , $2 \leq l \leq 3$, denotes the coordinate directions defining the transverse solution plane, and \bar{n}_l is the outward unit normal vector. In Eq. (1) s_i is a source or sink term for q_i and κ_i is the diffusion coefficient. The constants $a_{n,l}$ in Eq. (2) allow the specification of the appropriate boundary conditions on each boundary of the solution domain. The coefficients c_i and d_i as well as parametric or functional representations of the diffusion coefficients and source terms are supplied by the user as part of the input process and by providing additional coding as necessary.

The turbulent PNS equations and modelled equations described in the first paragraph are a subset of the general equation system and appropriate specifications of the coefficients, functionals, and source terms have been programmed. Nevertheless, there remains considerable latitude for the user, through the use of parameters specified on input, to further tailor the equation set for specific applications, especially with regard to the treatment of source and diffusion terms. Several of the original parameters are not described in Refs. [2-3] and some additional choices have been added by this author. The full equation set in a hybrid, as coded, form is included as Appendix A of this report and should serve to clarify and complete the descriptions in the above references.

The numerical solution of the governing equation set is accomplished by means of a finite-element algorithm. The particular algorithm employed by TWAKE is derived in Chapter 4 of Ref. [4] using the Galerkin-Weighted Residuals formulation. A brief outline of the algorithm is useful to introduce necessary terminology and for reference in subsequent sections of this report.

To a degree the finite-element algorithm is an integral transformation, transforming an initial value partial differential equation into a larger order system of ordinary differential equations. The algorithm is established by first sub-dividing the flow domain into a number of sub-domains or finite elements. Each element is associated with a number of discrete nodes located on (or within) the element boundary. The spatial variation within the element of any flow variable, q_e , is assumed in terms of interpolation polynomials, N , and the values (unknown) of the variable at the node points, Q , for example,

$$q_e(x_1, x_l) = \{N_k(x_l)\}^T \{Q(x_1)\}_e. \quad (4)$$

The curly braces denote column matrices whose order is equal to the number of nodes comprising the element and the elements of $\{N_k\}$ are polynomial functions complete to degree k . The numerically determined finite element approximation q^h , to the true solution q , is the summation of the independent solutions q_e ,

$$q(x_1, x_l) \approx q^h(x_1, x_l) \equiv \sum_{e=1}^M q_e(x_1, x_l), \quad (5)$$

where M equals the number of finite elements in the discretization. Within the Galerkin formulation, for example, the algorithm statement for a two-dimensional high Reynolds number boundary layer form of Eq. (A-1) with homogeneous boundary conditions is

$$\begin{aligned} S_e \left[\int_{R_e} \{N_k\} L(u_1^h) d\tau \right] &= S_e \left[\{U_1\}^T \int_{R_e} \{N_k\} \{N_k\} \{N_k\}^T dx_2 \{U_1\}' \right. \\ &+ \{Nu\}_e^T \int_{R_e} \{N_k\} \frac{d}{dx_2} \{N_k\} \frac{d}{dx_2} \{N_k\}^T dx_2 \{U_1\}_e + \{U_2\}_e^T \int_{R_e} \{N_k\} \{N_k\} \frac{d}{dx_2} \{N_k\}^T dx_2 \{U_1\}_e \\ &\left. + P'_\infty \int_{R_e} \{N_k\} dx_2 \right] \equiv \{0\}. \end{aligned} \quad (6)$$

In Eq. (6) S_e is the standard finite-element assembly operator, the "prime" denotes ordinary differentiation with respect to x_1 , and the Green-Gauss theorem has been used to transform the diffusion term. Each of the integrals are analytically evaluable upon specification of the interpolation polynomials in Eq. (4). The elements of the resultant hyper-matrices are themselves column or square matrices. Following the notation developed in Ref. [4], Eq. (6) is written

$$S_e \left[\{U_1\}_e^T [A3000] \{U_1\}'_e + (\{Nu\}_e^T [A3011] + \{U_2\}_e^T [A3001]) \{U_1\}_e + P_\infty \{A10\} \right] \equiv \{0\}. \quad (7)$$

The meaning of the matrix notation is easily deduced with reference to Eq. (6). For instance, $[A3001]$ in the convective term denotes that it is the matrix formed by integrating over a one-dimensional element (A), the product of 3 interpolation polynomial matrices (3), the first two of which are not differentiated with the third being differentiated (001). Matrices appropriate for the PNS equations have been coded into TWAKE for linear interpolation polynomials only ($k = 1$ in Eq. (4)), for both one-dimensional $[A...]$ elements and two-dimensional triangular $[B...]$ elements.

Application of the finite-element algorithm to each of the equations in Appendix A results in a system of ordinary differential equations (and algebraic) which are compactly written as.

$$[C_i]_e \{Q_i\}'_e + [D_i]_e \{Q_i\}_e + \{S_i\}_e = \{0\}. \quad (8)$$

where $\{S_i\}_e$ contains all of the non-homogeneous terms and $\{D_i\}$ contains the combined effects of convection and diffusion. To solve these equations the family of one-step implicit finite-difference integration algorithms is used,

$$\{F\}_{j+1} = \{Q_i\}_{j+1} - \{Q_i\}_j - \Delta x_1 \left(\theta \{Q_i\}'_{j+1} + (1 - \theta) \{Q_i\}'_j \right) = \{0\}, \quad (9)$$

where j is the x_1 step index, Δx_1 is the step-size and θ is the implicitness factor, with $\theta = \frac{1}{2}$ yielding the trapezoidal rule. Combining Eqs. (9) and (8) to eliminate the derivative yields the non-linear algebraic equation set,

$$\{F(\{Q_i\}_{j+1})\} = \{0\}. \quad (10)$$

The Newton iteration algorithm is applied to Eq. (10)

$$\{\delta Q_i\}_{j+1}^{p+1} = \{Q_i\}_{j+1}^{p+1} - \{Q_i\}_{j+1}^p = - \frac{\{F(\{Q_i\}_{j+1}^p)\}}{[J(\{Q_i\}_{j+1}^p)]}. \quad (11)$$

where p is the iteration index and $[J]$ is the Jacobian

$$[J(\{Q_i\})] \equiv \frac{\partial \{F\}}{\partial \{Q_i\}} \quad (12)$$

The equations to be solved at each iteration are Eq. (11) written in the form,

$$[J(\{Q_i\}_{j+1}^p)] \{\delta Q_i\}_{j+1}^{p+1} = - \{F(\{Q_i\}_{j+1}^p)\}, \quad (13)$$

where $\{\delta Q_i\}$, the iteration vector, is the dependent variable. The right side of the above equation is Eq. (10) evaluated at the p^{th} iteration,

$$\{F\}_{j+1}^p = [C_i] \left(\{Q_i\}_{j+1}^p - \{Q_i\}_j \right) + \Delta x_1 \left(\theta \{G_i\}_{j+1}^p + (1 - \theta) \{G_i\}_j \right). \quad (14)$$

where

$$\{G_i\}^p = [D_i] \{Q_i\}^p + \{S_i\}^p. \quad (15)$$

Iterating the solution until $\{F\}$ vanishes to within a defined tolerance implies the approximate vanishing of the iteration variable $\{\delta Q_i\}$ in Eq. (13) and hence convergence. The matrix solution technique for Eq. (13) is by L-U decomposition and back substitution.

The code has no "hard-coded" and linked control sequence until full initialization has occurred, and alternate paths exist in execution as well. The user is thus required to construct a problem-specific command sequence which will

- 1) generate the computational space,
- 2) specify the dependent variable set,
- 3) insure that fluid is initialized properly,
- 4) specify appropriate boundary conditions for each dependent variable,
- 5) specify certain parameters which control the flow path through the execution module and,
- 6) choose among several output options or otherwise provide output routines.

This has been done in Refs. [2-3] for a class of semi-bounded turbulent flows with an irregular computational domain. The next section and major portion of the remainder of this report will explain an input sequence appropriate for the simulation of surface ship wake flows. Comparing the two should efficiently educate the user in the machinations of operating the code and yield a versatility to create additional data sets. Once a command structure is developed for a class of flow geometries, the majority of the previous discussion in this section as well as other characteristics of the program can remain entirely transparent to the user. Since it is likely, however, that improved modelled equations or constitutive relations or more efficient solvers and iterative techniques will become available, an outline of the relevant sections of code which might be modified will be given in Section 4.

3. DESCRIPTION OF COMMAND DATA INPUT

The primary requirement of the user of the program is to construct an input sequence of commands which will describe the flow to be numerically simulated. Such a data sequence is included as Appendix A of Ref. [2] for the PNS computation of wing-body juncture flow. Another and quite different input sequence is included as Appendix B of this report for the turbulent wake behind a self-propelled surface ship. Following a few general remarks describing the command data structure, the next sub-section of this report will explain the functions of the fundamental commands included in Appendix B and indicate, where appropriate, how the data should be altered to compute additional flow fields of the same type. This will be followed by a short sub-section containing example modifications which either allow effects not included in the Appendix B commands or significantly alter the geometry. It will be assumed at this point that the reader is familiar with the texts of Appendices A and B.

The procedures for describing the flow-field geometry, initial fluid state and other characteristics pertinent to specific flows via the input sequence are quite flexible. At the core of the input module is subroutine **BDINPT**. This subroutine sequentially reads data card images from logical unit 5 (LU5). The individual card images must have the format (A8.72A1), the first eight characters (left-justified) of which contain a pre-programmed command name, while the remaining characters contain any additional parameters required by the specific command. The subroutines which read these card images translate the literal data into integer or real data as indicated. The commands can be categorized into two general groups, those which operate on and/or store numerical data, and those which cause control to transfer to other ensembles of subroutines which may themselves require additional literal or numerical data from LU5 or other logical units. The command card images are terminated by "T" if the data are numerical and by "DONE" if the data are literal. Upon completion of an operation specified by a command, control returns to **BDINPT** which reads the next card image. A list of available commands and their functions can be found in Appendix C of Ref. [2].

Immediately upon execution of **TWAKE**, control resides in program **MAIN**. The principal functions of this routine are to specify the size, **IZSIZE**, of the primary array in the code, **IZ(IZSIZE)**, and to call subroutine **BDINPT**. Currently **IZSIZE** = 700000 which is sufficient to compute a cross-plane containing 51x51 nodes. Additional code has been added to read file names (from LU5) and open files associated with logical units required for customized input and output subroutines. On some computer systems, these logical units may be designated by way of the job control language and in that event those particular statements in program **MAIN** should be deactivated. The first several card images in the data deck of Appendix B contain names for these additional LU's. It is not necessary to delete these card images if **MAIN** has been altered to attach these files via job control statements. This introduces an important

point regarding the command data structure. If a card image contains a character string which is not recognized as a legitimate command, then control continuously passes to subsequent images until a viable command is detected. Thus the entirety of the annotated data set in Appendix B (or even this entire report) can be read into the code verbatim.

Discussion of Appendix B

The first sensible statement to BDINPT in Appendix B is the command FENAME and this command must be the first. The command calls subroutine FENAME which initializes default entries in the integer and scalar arrays, IARRAY(500) and RARRAY(500), respectively. The specifications given these entries in the current version of FENAME are appropriate for the surface ship wake calculations of Refs. [6-7] and, in principle, the several subsequent commands in Appendix B of the IARRAY and RARRAY type need not be present. They have been included to remind the user of those entries which would most likely be altered to specify a different scaling to a geometrically similar flow. As is apparent from Appendix B a command of the type

```
IARRAY    2  7  T
```

puts the value of 7 into the second location of the array IARRAY. The comments in Appendix B suffice to explain the meaning and use of the several parameters which are explicitly referenced. Descriptions of most of the other defaulted entries in these arrays can be found in Refs. [2-3]. They need to be changed only if the user needs to significantly alter the flow type and geometry. Note the specification of the coefficients of the governing equations in the form presented in Appendix A. Note also that the last several of these array specification commands, along with the ICOND and EXIT commands, are preceded by the letter "C" and consequently they have no meaning to BDINPT. Removing the extraneous letter and left-justifying, however, will cause those particular elements of IARRAY to be assigned non-zero values and thus activate debug output statements (LU6) contained in the majority of the subroutines in the code. Some entries are assigned values greater than 1 because those particular integers are decremented by 1 each time output is requested. The ICOND command causes a print to LU6 of the current status of all non-zero entries in IARRAY and RARRAY and the EXIT will terminate execution when encountered. Obviously these parameters can be activated ($I > 0$) and deactivated ($I \leq 0$) at strategic locations in the input data sequence. There does exist a NAMELIST option in FENAME whereby the several parameter specifications in Appendix B can be accomplished by direct reference to their Fortran variable names. The option has not been used in Appendix B since, for illustrative purposes, it is convenient to isolate the parameter specifications in groups according to function. Most of the more widely used entries in RARRAY and IARRAY are equivalenced to local Fortran variable names in FENAME. The local variable names have been assembled into two NAMELIST data strings, one for each of the arrays. The parameter specification in Appendix B can as well be accomplished by:

```
IARRAY    500  1  T NAMELIST on
FENAME    T
&NAMEO1
  NTYPE=7, N2WAKE=1, NM=3, KROW=19, LCOL=19, NODE=400
&END
&NAMEO2
  UINF=6.7556, RHOINF=1.935, XMUINF=21.1E-6, REFL=1., TO=10.,
  TD=20., HSINIT=.01, HMAX=10., DELP=101.
&END
```

The FEDIMN command must occur after the grid-size parameters (NM, LCOL, KROW, NODE in Appendix B and others defaulted in FENAME) are prescribed. This subroutine activates the linear finite-element versions of the [A...] and [B...] matrices (depending on NM) discussed in Section 1 and loads default entries into several arrays. The critical function of FEDIMN, however, is to partition the IZ(IZSIZE) array. The macro-structure of this array is discussed in Ref. [3] and, due to its critical importance to creating new or altering existing code, Tables 1 and 2 are included to describe relevant further partitioning. It is sufficient at present to point out that the array always appears as equivalent to its real counterpart, RZ(IZSIZE), and every major column matrix in the code, whether integer or real, is a subset of this array. Subroutine FEDIMN calculates addresses (entry points in IZ/RZ) for these matrices and groups of

these matrices and stores these addresses in the first 200 locations of the IZ array. As an example, the following series of Fortran statements places the value of the x_2 space coordinate at computational node I into the local variable Z :

```
COMMON / ARRAYS / IZ(700000)
DIMENSION RZ(700000), L(200)
EQUIVALENCE ( IZ(1), RZ(1), L(1) )
EQUIVALENCE ( L(90), IX2COR )
Z = RZ(IX2COR + I - 1)
```

If the debug parameter, IARRAY(61), is active upon encountering FEDIMN, these addresses will be printed. The remainder of the input sequence is devoted to initializing (and sometimes further partitioning) the IZ array.

The next group of command data which deserve comments in addition to those given in the Appendix B annotation is the LINK4 sub-group. These commands, terminating with DONE, construct the finite-element domain. Commands VX2SCL and VX1SCL determine node spacing in the vertical and lateral directions, respectively, by using geometric progression. The card image immediately following VX2SCL, for example, contains 2 strings of numbers, separated by a comma for clarity (not necessary), which specify parameters for generating the vertical spacing,

```
VX2SCL      T
0.  9 -1.5 1.  , 9 -3.  1.  T
```

In this case the sets of numbers specify 2 super-elements in the vertical direction such that the first super-element begins at $x_2 = 0.0$, contains 9 finite elements (10 nodes over the span), and spans that axis until $x_2 = -1.5$, with node spacing determined by a progression ratio (Pr) of 1.0. The second super-element also contains 9 elements and continues the construction until $x_2 = -3.0$ with $Pr = 1.0$. Note that the command assumes subsequent super-elements begin where the prior super-element ended (-1.5 in this case) and the starting position is omitted. Since $Pr = 1.0$ (equi-spaced nodes) this construction could have identically been specified by

```
0.  18. -3.  1.  T
```

and the first format has been used to remind the user of the capability. If $Pr < 1.0$ the spacing becomes increasingly denser as the interval is spanned, for example, each of the constructions

```
0.  8 -1.5 1.25 , 12 -3.  1.  T
-3.  11 -1.  1.25 , 7 0.  .8 T
```

spans the same domain (originating at different points) and will cause a relatively denser spacing near $x_2 = 0.0$. Similar comments apply to the VX1SCL command. The call to ELEM constructs the finite-element network and node-connectivity table using the beginning and ending nodes specified by the NDECRD command, nodes 1 and 19 in each direction, respectively. A schematic of the grid formed for this geometry specification is given in Fig. 1. Note that since the reference length, IARRAY(43), has been assigned the value of 1.0, the space coordinates correspond directly to ft .

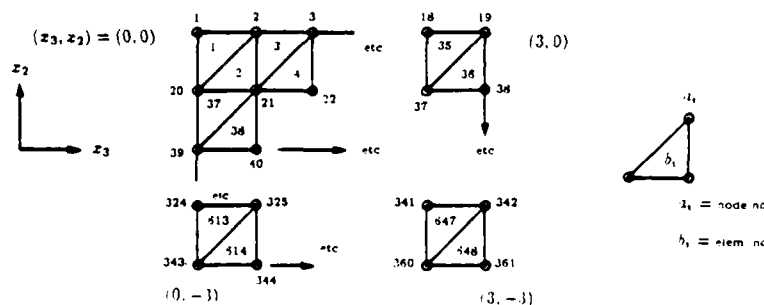


Fig. 1 - Schematic of finite-element domain

Each of the dependent variables has associated with it an integer number in accord with the table given in Appendix B. The KBNO sub-group of commands specifies boundary conditions to these variables according to their respective integer designations. The default boundary condition is vanishing gradient normal to the boundary in question. In Appendix B, for instance, there is no KBNO command for dependent variable 5 (turbulence kinetic energy) and thus either $\partial k / \partial x_1 = 0.0$ (left and right) or $\partial k / \partial x_2 = 0.0$ (bottom and top) as required. The domain created by the ELEM command above corresponds to the free surface wake experiments described in Ref. [7]. The top boundary is assumed to be the mean free surface in the "rigid-lid" approximation, the left boundary is the symmetry plane separating the port and starboard propellers, and the bottom and right boundaries are assumed to be free of viscous influence. If the initial (or boundary) conditions are asymmetric from starboard to port then a full planar solution is necessary. In this case, for instance, the KBNO 3 command and its associated data card image would be deleted to permit the lateral velocity, u_3 , to float on all boundaries. The data for ϕ (and p_p) would be replaced by,

```
KBNO      9 T   PHI BC: only free surface is impermeable
LEFT      BOTTOM      2      RIGHT      2
DONE
```

permitting entrainment at all boundary planes except the free surface. The "2's" in the above data cause the boundary node lists for the bottom and right boundaries to begin at the second node (proceeding counter-clockwise) thus preventing the bounding of the same node twice.

The next set of commands in Appendix B beginning with DESCRIPT 204 and continuing through VX3ST are concerned with specifying type of and format for printed output on LU6. The discussions in Refs. [2-3] are adequate to explain the general functions of these statements. Until the command IOSAVE is encountered the data are merely prescribing headers for scalar output and designating which scalars to write. The DESCRIPT 203 command provides header information for vector output. There is a one-to-one correspondence between the listed headers and the column vectors denoted by the code numbers listed under the IOSAVE command. The IOMULT command provides scalar multipliers for these 14 vectors and in this case the first 11 are multiplied by the scalar in RARRAY(2) (which happens to be 1.0), the next by RARRAY(171); and the last 2 again by RARRAY(2). These particular multipliers cause the output variables to be printed in their non-dimensional form as described in Appendix A. The particular pressure is multiplied by RARRAY(171) because the code operates with the variable divided by that constant. As an example to illustrate a couple of points, if the user wants to view the dimensional axial velocity and further wishes to record the effective turbulence diffusion coefficient in a form which compares it to the laminar viscosity (see Eqs. (A-1), (A-16) and (A-17)), then the last header card under DESCRIPT 203 might be altered to;

```
U3'U3'      PP      P      PHI      NUTRB/NU
```

the last statement under IOSAVE should be altered to;

```
8248 9248 1247 T
```

and the IOMULT data card should be altered to;

```
27 11*2 171 2*2 47, 15*1 T
```

These alterations, in addition to designating the header, cause the non-dimensional velocity variable defined in Appendix A to be multiplied by RARRAY(27) (u_∞ specified in a prior command in Appendix B) before printing. The multiplication of ν_t (variable number 1247) by RARRAY(47) (reference Reynolds number calculated during the input sequence) produces the ratio of the turbulent and laminar diffusion in Eq. (A-1). As to the construction of the composite numbers, the material in Ref. [2] on pages 21 and 35 is contradictory with that on page 21 correct. All arrays can be accessed using the correct procedures of Ref. [2] and appropriate addresses from the table on pages 55-58 of Ref. [3] and Tables 1 and 2 of this report. As is explained in Appendix B the axial locations where LU6 output will occur are specified by the VPVSX and VX3ST commands. Further comments regarding output options including customized output routines will be reserved for Section 4.

The LINK commands form the final sub-group which will complete initialization of the fluid. Commands of this type invoke a call to the named subroutine which in turn accesses other modules to perform operations according to the argument. The command,

LINK3 4 T

for instance, as part of the operations in subroutine LINK3 when the argument is 4, places calls to subroutines DIMEN and INDEX. Subroutine DIMEN calculates several non-dimensional parameters for later use during execution as well as initializes the turbulence model constants used in the equations of Appendix A. The call to INDEX finishes the partitioning of the IZ array begun in FEDIMN. It is here that several of the more commonly used addresses (entry points in the IZ array) are given special storage locations so that they do not have to be repeatedly calculated. For example the space between IZ(48) and IZ(49), allocated in FEDIMN, is further partitioned into segments of length $NYI \cdot NODE$ ($NYI = IARRAY(90)$). The starting address of each segment is stored in the common block JADRES which in most usages in the code has the form:

COMMON / JADRES / JU1 , JU2 , JU3 , JH , JK , JEPS , JPP , JP , JPHI , JEX(21)

In the execution module the current values of the dependent variables can be accessed by reference to the appropriate element of JADRES. For instance, the value of the non-dimensional u_1 velocity at node I is at $RZ(JU1+I-1)$. Other addresses, which have a one-to-one correspondence with terms or groups of terms in Eqs. (9) through (15), are calculated and stored in the common block DERIV. It should be noted here that those users who wish to modify code in the execution module will find a knowledge of the code in subroutines FEDIMN and INDEX essential to that task. Descriptions of the primary elements of DERIV can be found in Table 2.

The next commands, calls to subroutines GEOMFL and NODELM, complete the development of element matrices and vectors that are determinable solely from geometric data. It is in GEOMFL, for example, that the derivative matrix [A3011] of Eq. (7) or its 2D-element counterparts [B30112] and [B30113] are calculated.

The remaining task to be completed before entering the execution module is to initialize the fluid state at the initial plane $x_1 = T_0$. Inspection of the equation set in Appendix A shows that, at the minimum, non-zero and positive starting values of u_1 , k , and ϵ must be supplied on all nodes of the discretization. For swirling wake flows initial values of u_2 and u_3 should also be supplied. The user will likely find it convenient to develop his own input routines to initialize the dependent variable set. Examples of such specialized routines are the subroutines BLSIUS, BRDSHW, CRNINP, JNCINP, EDGINP, WAKPRO, VPIDATA, and DTNSRDC. These routines establish a variety of two and three-dimensional bounded and free-boundary flow fields from either empirical formulae or by accessing pre-processed data sets. The principal requirements in routines of this type are that the ARRAYS and JADRES common blocks be included and that the dependent variables ultimately be specified according to the node ordering determined during grid generation, the ELEM sub-group of commands discussed previously. While this latter point is obvious, consider that an alternate coordinate specification to that used in Appendix B, but one that yields precisely the identical geometry, is

```
VX2SCL      T
-3. 18.  0.  1.  T
VX1SCL      T
 3. 18.  0.  1.  T
```

This specification will reverse the node sequencing depicted in Fig. 1. If the user is uncertain as to how the elements and nodes are numbered, activation of the debug parameters prior to the ELEM command will generate a listing. In this application ($NTYPE = 7$) the LINK2 10 command accesses subroutine DTNSRDC which will read free-format data from LU19. The local file designation for LU19 is given as INPUT19 at the beginning of Appendix B and this data should be prepared in advance. In this case the first record in INPUT19 is the characteristic dissipation length scale, l_4 . Each succeeding record consists of 4 entries corresponding to the starting values of the dependent variables (u_1 , k , u_2 , u_3) at each node of Fig. 1 in sequence. The subroutine then initializes the dissipation function, ϵ , according to the discussion of Ref. [7]. The nodal data is an interpolation to the computational plane of the experimental data of Ref. [9]. The final two commands, LINK5 6 and LINK5 4, cause the initialization of the laminar

and turbulent diffusion coefficients and the Reynolds stresses according to the formulation in Appendix A. Note the presence in Appendix B of several commands preceding the transfer to execution (QKNINT) which are negated by the letter "C". Removing the extraneous letter will invoke a complete planar output of the geometry and fluid initialization for inspection prior to execution.

At this point initialization is complete and the annotated data set of Appendix B should be viewed as the general structure necessary for computation of surface wake flows. The following is a list of the items in that data set which should be changed to simulate a different flow of the same general class:

- 1) change the RARRAY entries corresponding to the fluid properties (UINF, RHOINF, XMUINF).
- 2) change the RARRAY entries corresponding to the longitudinal boundaries of the computational domain (TD, TD).
- 3) if there are to be more than 19 rows or 19 columns in solution, change the IARRAY entries for KROW and LCOL to be at least as large as the number of rows and columns, and change NODE to be at least $KROW * LCOL + 1$.
- 4) change the card images following the VX2SCL, VX1SCL, and NDECRD commands to reflect the new geometry.
- 5) change the KBNO sub-group of commands to bound the variables consistent with the new geometry specification.
- 6) change the card image following the VX3ST command to designate the longitudinal stations for complete planar output, and
- 7) change the data on LU19 to reflect the initial fluid state consistent with the new geometry or otherwise supply a subroutine similar to DTNSRDC.

Before leaving the discussion of command data set construction it is useful to illustrate by way of examples more general modifications necessary to calculate flow fields that deviate from the type heretofore considered. The following sub-section will briefly summarize modifications appropriate for a two-dimensional flow and for a flow with non-zero $\partial p_c / \partial x_1$. Studying the contrasts between the data sets (including that in Refs. [2-3]) will result in an increased versatility in using the code to calculate dissimilar flow fields.

Additional Data Sets

The hypothetical situation to be considered is a two-dimensional version of the Appendix B data. In this case it is supposed that the flow is invariant in the lateral (x_3) direction and that the initial fluid state is described by the data on the left column of nodes in Fig. 1. If the extent of the computational domain is the same, the only necessary change to the IARRAY and RARRAY specifications (or NAMELIST) is to set $NM = 2$. It is more efficient to alter LCOL and NODE to better fit the problem since FEDIMN will then allocate smaller blocks when partitioning RZ.

The IPINT vector should be altered to delete the integration of u_3 , e.g.,

```
IPINT      T
1 5 6 2 7 9 8 0 0 0 , 1 -2 -2 -5 6*0 , 10*I1 1 T
```

For the same span in x_2 , the only necessary changes to the ELEM sub-group of commands are to delete the VX1SCL command (and its associated data) and alter the NDECRD command and data to the single statement,

```
NDECRD      -1 T -1 indicates 2D grid
```

The IBORD command and data should be replaced by,

```
LINK1      2 T call FINDBE: examine geometry for boundary elements
```

The boundary conditions (KBNO) should be altered to delete the condition on u_3 and to remove the references to the RIGHT boundary for p_p and ϕ .

The remaining required change concerns reading the initial fluid state. Obviously subroutine DTNSRDC could be altered to store the first column of the current INPUT19 data (u_1, k, ϵ only) into the RZ array, or a modified data file could be pre-processed to be accessed by the existing subroutine. An alternative which illustrates further capabilities of the code is to delete altogether the link to TBLINP and DTNSRDC and add:

```

DEPVAR  1 2 T load U1
1.01225 1.01193 1.00424 1.01459 1.03879 1.02784 1.01261
1.01261 1.00843 1.00211 1.00053 9*1.0 T
RARRAY  2 .01 T set RARRAY(2)=.01
DEPVAR  5 2 T load TKE
.24124 .24795 .30290 .28394 .25587 .14605 .0430
.01045 11*1.0 T
RARRAY  2 .001 T set RARRAY(2)=.001
DEPVAR  6 2 T load EPS
.20312 .21166 .28578 .25937 .22188 .09568 .01529
.00183 11*1.0 T
RARRAY  2 1. T set RARRAY(2) back to 1.0

```

The DEPVAR N1 N2 command loads the RZ array beginning at entry points corresponding to the dependent variable specified by N1 (see the table in Appendix B). In this example each indicated variable will take on the 19 values specified by their respective data statements, with each datum multiplied by the value in RARRAY(N2). Note that the free format allows a short-hand notation for repeated data and the successive redesignation of RARRAY(2) disposes of the necessity to input the leading zeros.

The data set in Appendix B of Ref. [3] serves as an example of procedures required to include non-zero $\partial p_c / \partial x_1$ into a simulation. The explanation given in that reference, however, is extremely abstruse concerning the complementary (inviscid) pressure determination and only detailed deciphering of pressure-related subroutines along with subroutine DERVBL (conservation equation assembly) reveals how to and in what form to communicate the gradient information into the execution module. The fundamental requirement is to load the pressure data into the RZ array at locations IZ(105+I-1) where I ranges over the number of nodes in solution. Subroutine DERVBL expects the pressure gradient data in the form:

$$\frac{\partial P}{\partial x_1} = \left(\frac{\rho_\infty u_\infty^2}{p_\infty + \frac{1}{2} \rho_\infty u_\infty^2} \right) \frac{\partial}{\partial x_1} \left(\frac{p_c}{\rho_\infty u_\infty^2} \right), \quad (16)$$

and the user should note that the coefficient is accessible as the inverse of RARRAY(171). Examples of routines which accomplish this formulation for specific flow geometries are subroutines PRSGRD, GETPPR and JNCPFR. The first of these calculates the pressure term for an internal (rectangular duct) flow by global mass conservation and the latter two compute the term by interpolation of prescribed axial variations of $p_c(x_1)$. With the above as background the following word description will help clarify the operations performed by the several commands on pp. 91-92 of Ref. [3]. The LINK2 23 command reads and stores into temporary storage the data which immediately follows that command. These data represent 14 groups of $[x_p, C_p(x_3)_{x_1=x_p}]$ for the geometry of Ref. [1] under the assumption of flow symmetry about the corner-bisector. During the link to subroutine MODPCP these data are used to specify pressure boundary conditions at each node under the KBNO 8 command (p. 90). A solution to Eq. (A-5) is then obtained for each of the 14 stations, the axial coordinates of which are now stored sequentially beginning at RZ(IZ(139)). The solutions are re-assembled into sub-groups of 14 values (solutions) per node and stored according to the normal node sequence beginning at RZ(IZ(140)). During execution subroutine MODPPR is called at the initiation of each marching step and the x_p interval bracketing the current x_1 determines entry points to the pressure table allowing evaluation of Eq. (16) by linear interpolation. These results are loaded sequentially, beginning at RZ(IZ(105)), for transfer to DERVBL.

As a final example which simply illustrates details of the above description, consider that the zero pressure gradient calculation of Appendix B has been established and one wishes to account for the resultant axial gradients of p_p in a subsequent pass through the flow field. In this subsequent iteration, the prior distribution of p_p will be identified with the current p_c . It is assumed that on the first pass the p_p distributions were saved (unformatted) on LU11 in the form specified in subroutine PLTOUT. For purposes of illustration it is further assumed that these distributions were stored only at the axial positions specified under VX3ST in Appendix B, however, it is in the near wake that the procedure has

more relevance. For these conditions the following additional parameter specifications should be made before the call to FEDIMN:

```
IARRAY  405    1 T (IPRSCAL) turn on pressure table look-up procedures
IARRAY  394    4 T (NPOTAB) 4 tables of pressure coefficients
IARRAY  371 361 T (NPVSXT) 361 values in each table
IARRAY  161 1444 T (NPVSI) storage allocation 4*361 entries
```

The VPVSI command in Appendix B is removed and the following series of statements should follow the VX3ST specification:

```
RARRAY  450 2. T temporarily set RARRAY(450)=2.
READ    11 161 74 T
SETVAL  1444 74 74 450 T
LINK2   23 T call CPSTUP to create tables
LINK1   11 T call NODPPR to initialize p grad.
RARRAY  450 0. 500 T reset RARRAY(450)=0.
```

The READ command reads from LU11 a vector of length IARRAY(161) and stores it temporarily beginning at RZ(IZ(74)). Since subroutine CPSTUP expects data in pressure coefficient form, the SETVAL statement multiplies the vector by RARRAY(450) and places the result in the same temporary storage location. The call to CPSTUP creates the tables and finally NODPPR initializes p_c and $\partial p_c / \partial x_1$.

The following section will give a brief overview of the macro-structure of the execution module as it applies to wake flows.

4. OVERVIEW OF THE EXECUTION MODULE

The QKNINT command near the end of the Appendix B data transfers control into the main drive loop. A flow chart for the routines performing the principal operations during execution is shown in Fig. 2. The flow paths branching to the sides of QKNINT pertain to output options and will be discussed at the end of this section. The following several paragraphs will summarize the major functions of the remaining subroutines. Tables 1 and 2 contain additional information regarding the RZ array entry points for those users having a need for detailed code translation.

The call to IMPLCT marks the initiation of a step in the axial direction, corresponding to the j index of Eq. (9). At this point any necessary parametric evaluations should be performed, such as those necessary for variable geometry or non-zero pressure gradient. These types of subroutines are user supplied. Examples of routines which fill the column matrix for $\partial p_c / \partial x_1$ are subroutines JNCPPR, GETPPR, and PRSGRD. The first two of these compute gradients by interpolations of either prescribed data or solutions of Eq. (A-5). Subroutine PRSGRD calculates the pressure gradient in internal flows from global mass conservation considerations.

The sequence from DERVBL through STRF in Fig. 2 is the iteration loop signified by index p in Eq. (11). Subroutine DERVBL performs the finite-element assembly indicated in Eqs. (6) and (7). The user will recognize in DERVBL the application of the various RARRAY coefficients discussed in Appendix A and the ultimate assembly of Eqs. (10) and (14) for each parabolic equation. The master loop in subroutine IMPSLV is over the dependent variable index, NP , equivalent to the index i of the equations in Section 2. As NP increments from 1 to the total number of marching equations (5 in this case), Eqs. (13) are solved in the order specified by the IPINT command. Solution is by L-U decomposition and back substitution (BANDSL). The output from BANDSL is the iteration variable, δQ_i , forming the left side of Eq. (11). If for any variable at any solution node, n

$$\frac{(\delta Q_i)_n}{(Q_{max})_i} > R_{14},$$

convergence is not considered achieved (any NPCONV(NP) greater than zero) and Q_i^{p+1} is determined by Eq. (11). Currently R_{14} (RARRAY(14)) is set to 10^{-4} in FENAME. Following IMPSLV, subroutines SETDIF and RNLDST update the diffusion coefficients and Reynolds stresses. If the convergence test was failed on any variable, two successive passes through subroutines PPRES and STRF accomplish the solutions of

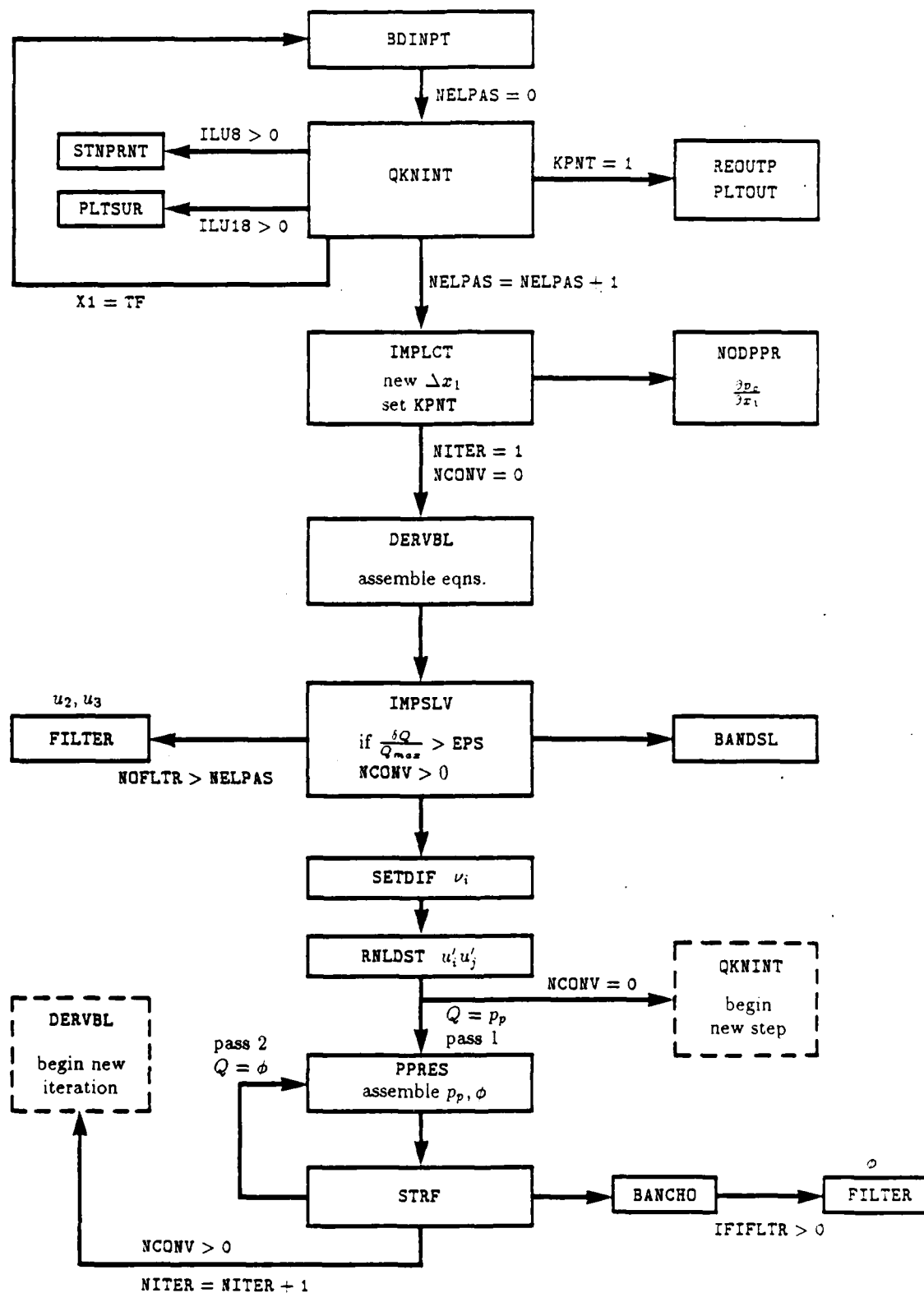


Fig. 2 - Flow chart of execution module

the Poisson equations for p_p and ϕ and control returns to DERVBL. If convergence was achieved for all variables (all NPCONV(NP) equal zero), control returns to QKNINT after the stress update. Flow charts of DERVBL and IMPSLV are given in Ref. [3].

The references to subroutine FILTER in Fig. 2 deserve special mention. Chapters 5 and 6 of Ref. [4] discuss the possibility of waves occurring in the cross-plane velocity component solutions for the linear basis algorithm ($k = 1$ in Eq. (4)). The mechanism for initiating these waves is the action of the continuity penalty function which for the linear cardinal basis enforces essentially a central difference approximation. If a calculation is exhibiting this behavior, simple grid refinement can exacerbate the problem. The original code contains options to either filter the secondary flow velocity variables directly, filter the iteration variable for the those variables, or apply no filter, with the former of these options implemented by default. This user has computed drag-wake solutions (Ref. [6]) using both "on" possibilities for the original filter, and observed only insignificant perturbations in the entrainment induced u_2, u_3 solutions. Operation with the filter ofldid indeed reveal the presence of small spurious waves in the ϕ solution and a resulting infection of the cross-plane velocities. It is emphasized that this observed behavior was small and sensibly not unstable at least for the initial conditions and longitudinal extent of the computations in that study. Nevertheless these observations along with the comments in Ref. [4] give reason to monitor and condition the solution if necessary. If there is a non-zero swirl component in the initial condition, however, application of the filter to the secondary flow velocities is inappropriate since the result is an immediate artificial smearing of those components. Grid refinement will alleviate the smearing effect at the expense of a corresponding increase in computer time. Consequently the current code has a filter option at subroutine STRF which will operate only on the penalty function. The default condition is an active ϕ filter set in FENAME by IARRAY(403)=1. The u_2, u_3 filters, controlled by the IARRAY entries (397), (401), (402) and (404) are currently off. The use of these filter options can easily be deduced by examining the code in subroutines IMPSLV and QKNINT.

Subroutine QKNINT transfers control to the output driver whenever the parameter IARRAY(86) (KPNT) is non-zero. This parameter is set on each pass through IMPLCT and will be non-zero

- 1) at values of the step index NELPAS which are integer multiples of KNTPAS (IARRAY(167)),
- 2) at the transverse planes closest to the positions defined by

$$\frac{x_1}{R_{35}} = \frac{nR_{13}}{100}$$

where n is an integer and the RARRAY entries are described in Appendix B, and

- 3) at planes closest to the positions listed via the VX3ST command in Appendix B.

If KPNT is non-zero full planar output (LU6) of all arrays specified with the IOSAVE command will result. The format for this output is flexible and will automatically be determined to resemble the problem geometry constructed during the ELEM sequence. At axial stations where REOUTP is accessed, subroutine PLTOUT writes the entire dependent variable set (and Reynolds stresses) to LU15 in a column matrix form corresponding to the node sequencing from grid construction. The nodal data is preceded by a sequential listing of the x_2, x_3 coordinates of each node, which for invariant geometry are written on the first pass only. If the additional parameters ILU18 (IARRAY(408)) and/or ILU8 (IARRAY(407)) are specified non-zero during initialization, then subroutines PLTSUR and/or STNPRNT will be called at each axial step. Subroutine PLTSUR writes to LU18 the full set of computed variables at each node in the horizontal plane $x_2 = 0$ (top row of Fig. 1), corresponding to the mean free surface. Currently a call to STNPRNT will initiate prints to each of the logical units 8, 9, 16 and 17. These are characteristic data of the calculation and are for the most part comprised of extrema of the several dependent variables in the local transverse plane. All potentially relevant IZ array entry points are included in these routines and they can quite easily be tailored to special needs. Sample data from all output units are included in Appendix C.

5. CLOSING REMARKS

The subroutines comprising TWAKE are grouped according to their primary functions and listed in Table 3. These modules, the input data files, and all output files resulting from the execution of the Appendix B data have been written to magnetic tape for delivery to OCNR (Code (12)). The data set has been executed on several NRL computers with identical results to 4 significant digits. The following is a tabulation of the execution times:

<u>SYSTEM</u>	<u>CPU-min.</u>
CRAY	2.24
VAX 11/785	35.17
VAX 11/780	41.83
HP 9000	215.95

6. ACKNOWLEDGEMENTS

The author wishes to thank Dr. E. W. Miner and Dr. G. A. Keramidas for providing valuable assistance in installing TWAKE on various computer systems.

7. REFERENCES

- [1] A. J. Baker, "The CMC:3DPNS Computer Program for Prediction of Three-Dimensional Subsonic, Turbulent Aerodynamic Juncture Region Flow: Volume I - Theoretical," NASA Contractor Report 3645, 1982.
- [2] P. D. Manhardt, "The CMC:3DPNS Computer Program for Prediction of Three-Dimensional Subsonic, Turbulent Aerodynamic Juncture Region Flow: Volume II - Users' Manual," NASA Contractor Report 165997, 1982.
- [3] J. A. Orzechowski, "The CMC:3DPNS Computer Program for Prediction of Three-Dimensional Subsonic, Turbulent Aerodynamic Juncture Region Flow: Volume III - Programmers' Manual," NASA Contractor Report 165998, 1982.
- [4] A. J. Baker, *Finite Element Computational Fluid Mechanics*. McGraw-Hill, Hemisphere, New York, 1983.
- [5] T. F. Swean, Jr. and G. A. Keramidas "The Turbulent Wake of a Flat Plate Near a Free Surface," Proc. of the Second International Conference on Computational Methods and Experimental Measurements, 1984.
- [6] T. F. Swean, Jr., "Calculations of the Turbulent Wake Behind a Slender Self-Propelled Double-Body and Comparisons with Experiment," NRL Memorandum Report 6075, 1987.
- [7] T. F. Swean, Jr., "Numerical Simulations of the Wake Downstream of a Twin-Screw Destroyer Model," NRL Memorandum Report 6131, 1987.
- [8] COSMIC Software Catalog: 1987 Edition, CSMIC, Univ. of Georgia, Athens, Georgia. NASA Contractor Report 179669, 1987.
- [9] W. T. Lindenmuth, private communication: data from the model experiments at DTRC - inboard propeller rotation, Jan. 1987.

TABLE 1 - Some important IZ array entry points

IZ entry	commonly used name	reference for node I	remarks
43	IC200	reference by element	standard matrices: see FEDIMN
48	IYY	RZ(IYY+NYI*(IPLOC-1)*NODE+I-1)	dependent variables ¹
71	ITEMP1	RZ(ITEMP1+(NS+1)*NODE+I-1)	Reynolds stresses ²
74	ITEMP4	RZ(ITEMP4+I-1)	RZ(NZF+I-1) in IMPSLV ⁴
89	IX1COR	RZ(IX1COR+I-1)	nodal x_3 coordinates
90	IX2COR	RZ(IX2COR+I-1)	nodal x_2 coordinates
91	IPRESS	RZ(IPRESS+I-1)	nodal values of p
92	IAMU	RZ(IAMU+I-1)	nodal laminar viscosity
103	IVEL	RZ(IVEL+I-1)	temporary storage, u_2
104	IW	RZ(IW+I-1)	temporary storage, u_3
105	IPRGRD	RZ(IPRGRD+I-1)	nodal $\partial p_c / \partial x_1$ term
109	IGEOM1	reference by element ³	natural coordinate derivative (x_3)
110	IGEOM2	reference by element ³	natural coordinate derivative (x_2)
136	IADIF	RZ(IADIF+I-1)	nodal values of v_i

Notes

- 1 IPLOC is the location in the IPINT vector. In Appendix B, for example, IPLOC=2 for the turbulence kinetic energy. NYI and NODE are entries 90 and 55 in IARRAY.
- 2 $NS = \{1, 2, 3, 4, 5, 6\}$ for $\{u'_1 u'_1, u'_1 u'_2, u'_1 u'_3, u'_2 u'_2, u'_2 u'_3, u'_3 u'_3\}$
- 3 See subroutine RNLDST for examples of usage.
- 4 For each variable NP in turn, NZF points to the left-hand-side of Eq. (14) (including penalty term for u_2, u_3) upon entering BANDSL. At exit from BANDSL, NZF points to left-hand-side of Eq. (11).

TABLE 2 - Some important addresses in COMMON / DERIV /

DERIV entry	commonly used name	reference for node I	remarks
25	IIYY	RZ(IIYY+NYI*(IPLOC-1)*NODE+I-1)	IIYY \equiv IZ(48)
36	NIYB	RZ(NIYB+NP+I-1)	dep. var. ¹ : $\{Q_i\}_{j+1}^p$ in Eq. (14)
37	NIZB	RZ(NIZB+NP+I-1)	$\{G_i\}_{j+1}^p$ in Eq. (14)
61	NIYY	RZ(NIYY+NP+I-1)	same as DERIV(36)
62	NIZZ	RZ(NIZZ+NP+I-1)	same as DERIV(37) ²
63	NIRU	RZ(NIRU+I)	JADRES(1)-1 ³
64	NIRV	RZ(NIRV+I)	JADRES(2)-1
65	NIRW	RZ(NIRW+I)	JADRES(3)-1
71	NIRE	RZ(NIRE+I)	JADRES(4)-1
72	NIRF	RZ(NIRF+I)	JADRES(5)-1
74	NIYYO	RZ(NIYYO+NP+I-1)	$\{Q_i\}_j$ in Eq. (14)
75	NIZZO	RZ(NIZZO+NP+I-1)	$\{G_i\}_j$ in Eq. (14)

Notes

- 1 NP $\equiv i$ in Eq. (1) for all usages in table. NP for a particular dependent variable is determined according to the variable's location in the IPINT vector.
- 2 RZ(NIZZ+NP+2*NODE+I-1) contains the first term on right side of Eq. (14).
- 3 JADRES(n) refers to n^{th} element of COMMON / JADRES /

TABLE 3 - TWAKE subroutines sorted according to primary function

<u>Input</u>	<u>Output</u>	<u>Utility</u>	<u>Eqn. Solve</u>
ADDDDEL	CALORD	GETDAT	ASSMAT
BDATA	COMOC	LINK1	ASMSQ
BDINPT	DRVBUG	LINK2	BANCHO
BLSIUS	FEPL0T	LINK3	BANDSET
BNDSET	DPSISQ	LINK4	BANDSL
BRDSHW	ICOND	LINK5	BCOND T
COLS	OUTNOD	LOCATE	CONTES
CPINIT	OUTPG	LOOK	DERVBL
CPSTUP	OUTVEC	LOOK1	DERVDX
CRNINP	OUTVEH	LOOKAV	DFCFBL
CSFINP	PBLANK	MATSUM	DFCFNS
DELADD	PLILNK	MINMAX	DRHOBL
DELELM	PLTOUT	MNMX	DUDY
DELETE	PLTSUR	NODPPR	EDGPPR
DELNOD	PRINTA	RECIP	FILTER
DESCRP	REOUTP	RESET	GETFSL
DIMEN	RITE	RESETI	GETPPR
DSCRTZ	SCALEV	EXIT	IMPLCT
DTNSRDC	SETSCL	SETRZS	IMPSLV
EDGINP	STNPRNT	SETVAL	JNCPPR
EDGSYM	STOUT1	TBLINP	NBNDRY
ELEM		TCHECK	PPRES
FEDIMN		TIMETK	PRSGRD
FENAME		TRAPIN	QKNINT
FINDBE		VARMAX	RNLDST
GEOMDR		VECMUL	SETDIF
GEOMFL		XYSICAL	SETIMP
GETALC		ZZZZZZ	STCODE
GETBCD			STRF
GETBCM			SUMKEY
GETBND			TAUW
INDEX			TRBTHK
NDECRD			WKPRES
MODEL M			WLFLXS
NODFIX			
NODPCP			
ORDER			
PRATIO			
READER			
READV1			
REDREL			
REORDR			
ROWS			
SETCNT			
TWPROT			
VPIDATA			
XYCRDM			

APPENDIX A - PROGRAMMED FORMS OF CONSERVATION EQUATIONS

The programmed conservation equations are non-dimensional, with all variables having been non-dimensionalized using reference values of length, viscosity, density and velocity as appropriate, for example,

$$x_1^* \equiv \frac{x_1}{L}, u_1^* \equiv \frac{\bar{u}_1}{u_\infty}, u_1' u_2'^* \equiv \frac{\overline{u_1' u_2'}}{u_\infty^2}, p^* \equiv \frac{\bar{p}}{\rho_\infty u_\infty^2}, k^* \equiv \frac{k}{u_\infty^2}.$$

The overbars indicate conventional time-averaging and the primes denote fluctuating (turbulence) components. Omitting the asterisks for convenience and assuming x_1, x_2, x_3 correspond to the streamwise, vertical, and lateral directions, respectively, the equations for an isothermal, constant property, incompressible flow are programmed in the following parameterized form:

streamwise momentum

$$u_1 \frac{\partial u_1}{\partial x_1} + R_{385} \left(u_2 \frac{\partial u_1}{\partial x_2} + u_3 \frac{\partial u_1}{\partial x_3} \right) - \frac{\partial}{\partial x_2} \left[(Re_L^{-1} + (1 - R_{281}) \nu_t) \frac{\partial u_1}{\partial x_2} \right] - \frac{\partial}{\partial x_3} \left[(Re_L^{-1} + (1 - R_{281}) \nu_t) \frac{\partial u_1}{\partial x_3} \right] + R_{281} \left(\frac{\partial}{\partial x_2} (u_1' u_2') + \frac{\partial}{\partial x_3} (u_1' u_3') \right) + \frac{\partial p_c}{\partial x_1} = 0, \quad (A-1)$$

vertical momentum

$$u_1 \frac{\partial}{\partial x_1} (u_2 - R_{353} \mathcal{F}(\partial \phi / \partial x_2)) + R_{348} \left(u_2 \frac{\partial u_2}{\partial x_2} + u_3 \frac{\partial u_2}{\partial x_3} \right) - \frac{\partial}{\partial x_2} \left[\left(\frac{R_{346}}{Re_L} + (1 - R_{282}) \nu_t \right) \frac{\partial u_2}{\partial x_2} \right] - \frac{\partial}{\partial x_3} \left[\left(\frac{R_{346}}{Re_L} + (1 - R_{282}) \nu_t \right) \frac{\partial u_2}{\partial x_3} \right] - \frac{\partial}{\partial x_3} \left(\frac{R_{346}}{Re_L} \frac{\partial u_3}{\partial x_2} \right) + R_{282} \left(\frac{\partial}{\partial x_2} (u_2' u_2') + \frac{\partial}{\partial x_3} (u_2' u_3') + R_{399} \frac{\partial}{\partial x_1} (u_1' u_2') \right) + R_{339} \frac{\partial p_c}{\partial x_2} + R_{340} \frac{\partial p_p}{\partial x_2} = 0, \quad (A-2)$$

lateral momentum

$$u_1 \frac{\partial}{\partial x_1} (u_3 - R_{353} \mathcal{F}(\partial \phi / \partial x_3)) + R_{348} \left(u_2 \frac{\partial u_3}{\partial x_2} + u_3 \frac{\partial u_3}{\partial x_3} \right) - \frac{\partial}{\partial x_2} \left[\left(\frac{R_{346}}{Re_L} + (1 - R_{282}) \nu_t \right) \frac{\partial u_3}{\partial x_2} \right] - \frac{\partial}{\partial x_3} \left[\left(\frac{R_{346}}{Re_L} + (1 - R_{282}) \nu_t \right) \frac{\partial u_3}{\partial x_3} \right] - \frac{\partial}{\partial x_2} \left(\frac{R_{346}}{Re_L} \frac{\partial u_2}{\partial x_3} \right) + R_{282} \left(\frac{\partial}{\partial x_2} (u_2' u_3') + \frac{\partial}{\partial x_3} (u_3' u_3') + R_{399} \frac{\partial}{\partial x_1} (u_1' u_3') \right) + R_{339} \frac{\partial p_c}{\partial x_3} + R_{340} \frac{\partial p_p}{\partial x_3} = 0, \quad (A-3)$$

continuity-penalty function

$$\frac{\partial^2 \phi}{\partial x_2^2} + \frac{\partial^2 \phi}{\partial x_3^2} - \frac{\partial u_1}{\partial x_1} - \frac{\partial u_2}{\partial x_2} - \frac{\partial u_3}{\partial x_3} = 0, \quad (A-4)$$

complementary pressure

$$\frac{\partial^2 p_c}{\partial x_2^2} + \frac{\partial^2 p_c}{\partial x_3^2} = 0. \quad (A-5)$$

particular pressure

$$\begin{aligned} & \frac{\partial^2 p_p}{\partial x_2^2} + \frac{\partial^2 p_p}{\partial x_3^2} + R_{396} \left[\frac{\partial}{\partial x_2} \left(\frac{\partial}{\partial x_2} (u'_2 u'_2) + \frac{\partial}{\partial x_3} (u'_2 u'_3) \right) + \frac{\partial}{\partial x_3} \left(\frac{\partial}{\partial x_2} (u'_2 u'_3) + \frac{\partial}{\partial x_3} (u'_3 u'_3) \right) \right] \\ & + R_{394} \left[\frac{\partial}{\partial x_2} \left(u_2 \frac{\partial u_2}{\partial x_2} + u_3 \frac{\partial u_2}{\partial x_3} + R_{398} u_1 \frac{\partial u_2}{\partial x_1} \right) + \frac{\partial}{\partial x_3} \left(u_2 \frac{\partial u_3}{\partial x_2} + u_3 \frac{\partial u_3}{\partial x_3} + R_{398} u_1 \frac{\partial u_3}{\partial x_1} \right) \right] = 0, \quad (A-6) \end{aligned}$$

where, $p = p_p + p_c$,

turbulence kinetic energy

$$\begin{aligned} & u_1 \frac{\partial k}{\partial x_1} + R_{384} \left(u_2 \frac{\partial k}{\partial x_2} + u_3 \frac{\partial k}{\partial x_3} \right) - \frac{\partial}{\partial x_2} \left[\left(Re_L^{-1} + (1 - R_{283}) \frac{\nu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_2} \right] \\ & - \frac{\partial}{\partial x_3} \left[\left(Re_L^{-1} + (1 - R_{283}) \frac{\nu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_3} \right] - \mathcal{P} + \epsilon \\ & + R_{283} C_k \left[\frac{\partial}{\partial x_2} \left(\frac{k}{\epsilon} \left(u'_2 u'_2 \frac{\partial k}{\partial x_2} + u'_2 u'_3 \frac{\partial k}{\partial x_3} \right) \right) + \frac{\partial}{\partial x_3} \left(\frac{k}{\epsilon} \left(u'_2 u'_3 \frac{\partial k}{\partial x_2} + u'_3 u'_3 \frac{\partial k}{\partial x_3} \right) \right) \right] = 0, \quad (A-7) \end{aligned}$$

isotropic dissipation function

$$\begin{aligned} & u_1 \frac{\partial \epsilon}{\partial x_1} + R_{384} \left(u_2 \frac{\partial \epsilon}{\partial x_2} + u_3 \frac{\partial \epsilon}{\partial x_3} \right) - (1 - R_{283}) \left[\frac{\partial \epsilon}{\partial x_2} \left(\frac{\nu_t}{\sigma_\epsilon} \frac{\partial \epsilon}{\partial x_2} \right) + \frac{\partial \epsilon}{\partial x_3} \left(\frac{\nu_t}{\sigma_\epsilon} \frac{\partial \epsilon}{\partial x_3} \right) \right] \\ & + R_{283} C_\epsilon \left[\frac{\partial}{\partial x_2} \left(\frac{k}{\epsilon} \left(u'_2 u'_2 \frac{\partial \epsilon}{\partial x_2} + u'_2 u'_3 \frac{\partial \epsilon}{\partial x_3} \right) \right) + \frac{\partial}{\partial x_3} \left(\frac{k}{\epsilon} \left(u'_2 u'_3 \frac{\partial \epsilon}{\partial x_2} + u'_3 u'_3 \frac{\partial \epsilon}{\partial x_3} \right) \right) \right] \\ & - C_{\epsilon_1} \mathcal{P} \frac{\epsilon}{k} + C_{\epsilon_2} \frac{\epsilon^2}{k} = 0, \quad (A-8) \end{aligned}$$

turbulence production .

$$\mathcal{P} = -u'_1 u'_2 \frac{\partial u_1}{\partial x_2} - u'_1 u'_3 \frac{\partial u_1}{\partial x_3} - u'_2 u'_3 \left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2} \right) - (u'_2 u'_2 - u'_1 u'_1) \frac{\partial u_2}{\partial x_2} - (u'_3 u'_3 - u'_1 u'_1) \frac{\partial u_3}{\partial x_3}, \quad (A-9)$$

kinematic turbulent stresses:

$$u'_1 u'_1 = C_1 k - R_{400} C_2 \nu_t \frac{k}{\epsilon} \left(\left(\frac{\partial u_1}{\partial x_2} \right)^2 + \left(\frac{\partial u_1}{\partial x_3} \right)^2 \right) - 2 R_{401} \nu_t \left(\frac{\partial u_1}{\partial x_1} \right), \quad (A-10)$$

$$u'_2 u'_2 = C_3 k - R_{402} C_2 \nu_t \frac{k}{\epsilon} \left(\frac{\partial u_1}{\partial x_2} \right)^2 - 2 R_{403} \nu_t \left(\frac{\partial u_2}{\partial x_2} \right), \quad (A-11)$$

$$u'_3 u'_3 = C_3 k - R_{404} C_2 \nu_t \frac{k}{\epsilon} \left(\frac{\partial u_1}{\partial x_3} \right)^2 - 2 R_{405} \nu_t \left(\frac{\partial u_3}{\partial x_3} \right), \quad (A-12)$$

$$u'_1 u'_2 = -\nu_t \left(\frac{\partial u_1}{\partial x_2} \right) - C_2 \nu_t \frac{k}{\epsilon} \left(R_{406} \frac{\partial u_1}{\partial x_3} \left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2} \right) - 2 R_{407} \frac{\partial u_1}{\partial x_2} \frac{\partial u_3}{\partial x_3} \right), \quad (A-13)$$

$$u'_1 u'_3 = -\nu_t \left(\frac{\partial u_1}{\partial x_3} \right) - C_2 \nu_t \frac{k}{\epsilon} \left(R_{408} \frac{\partial u_1}{\partial x_2} \left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2} \right) - 2 R_{409} \frac{\partial u_1}{\partial x_3} \frac{\partial u_2}{\partial x_2} \right), \quad (A-14)$$

$$u'_2 u'_3 = -\nu_t \left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2} \right) - R_{411} C_2 \nu_t \frac{k}{\epsilon} \left(\frac{\partial u_1}{\partial x_2} \frac{\partial u_1}{\partial x_3} \right), \quad (A-15)$$

The turbulent kinematic viscosity and the reference Reynolds number are given by.

$$\nu_t = C_4 \frac{k^2}{\epsilon}, \quad (\text{A-16})$$

and,

$$Re_L = \frac{u_\infty L}{\nu_\infty}. \quad (\text{A-17})$$

respectively. The function \mathcal{F} in the cross-plane momentum equations is included to illustrate the action of the penalty (continuity) constraint term in the numerical algorithm (see Ref. [4], chapter 7). Aside from the " R_{nnn} " coefficients, the model contains the 10 constants $\{C_1, C_2, C_3, C_4, \sigma_k, \sigma_\epsilon, C_{\epsilon_1}, C_{\epsilon_2}, C_k, C_\epsilon\}$ which take on the commonly accepted values $\{0.94, 0.067, 0.56, 0.068, 1.0, 1.3, 1.44, 1.92, 0.12, 0.09\}$. Most of these model constants are specified in subroutine FENAME, however a few are computed in subroutines DIMEN and DERVBL.

The subscripts on the R_{nnn} coefficients in the above equations refer to locations in the scalar array RARRAY(500). This array is initialized to default values in FENAME and modified to be problem specific during the input sequence. The coefficients have obvious utility in specifying a problem class from the general convp576section-diffusion (Eq. (1)). Restricting attention to the PNS equations, several of the coefficients, particularly those modifying the convection and source (pressure gradient) terms, are extraneous, and have utility only for debugging or sensitivity studies. Those coefficients modifying the eddy viscosity diffusion and Reynolds stress-gradient terms have more utility since they potentially provide an additional control over the degree of non-linearity and explicitness in the discretized equations by specifying more or less of the diffusion in the source terms. It should be noted, however, that in this version of the code, the Reynolds stresses are updated at every iteration, and as such any representation $0 \leq R_{nnn} \leq 1$ will give essentially identical results if only the lead order terms are retained in the Reynolds stress constitutive equations. For instance, specifying $R_{281} = 1$ or 0 in Eq. (A-1) yields the same result if only the first terms in Eqs. (A-13) and (A-14) are used and if the stresses are computed at the current iteration. In the original version of the code the following default values of coefficients in Eqs. (A-1) through (A-8) were set:

$$\begin{aligned} R_{385} = R_{348} = R_{282} = R_{339} = R_{340} = R_{396} = R_{394} = R_{384} &= 1, \\ R_{281} = R_{346} = R_{399} = R_{398} = R_{283} &= 0, \\ R_{353} &= -1. \end{aligned}$$

In the current version of the code the above specifications are retained with the exceptions that $R_{282} = 0$ and $R_{346} = 1$, thus all diffusion terms are determined to the same degree of approximation. Calculations were performed for a variety of permutations of these coefficients during the studies reported in Refs. [6,7] and differences in the results due to laminar diffusion or the alternate forms of Eqs. (A-2) and (A-3) were insignificant.

The stress specification in Eqs. (A-10) through (A-15) is different from the original specification by the presence of the second terms in the equations for $u'_1 u'_2$ and $u'_1 u'_3$. These were added so as to include the full model specified in Ref. [4]. All of the stress coefficients have default values of unity.

APPENDIX B - COMMAND INPUT DATA FOR WAKE SIMULATIONS

OUTPUT8
OUTPUT9
OUTPUT15
OUTPUT16
OUTPUT17
OUTPUT18
INPUT19

TWAKE DATA : DTNSRDC DESTROYER : INITIALIZE AT 10 FT.
INBOARD PROPELLER ROTATION

Call SUBROUTINE FENAME to initialize default IARRAY and RARRAY entries.

FENAME T SUBROUTINE BDINPT calls SUBROUTINE FENAME on this command.

Define the type of flow for simulation.

IARRAY 2 7 T sets IARRAY(2)=NTYPE=7

NTYPE controls subsequent calls within SUBROUTINE TBLINP and SUBROUTINE NODPPR. This value will cause the fluid state to be initialized according to DTNSRDC experimental data that has been pre-processed and saved in file INPUT19 above. This parameter also controls the flow path for computing the axial pressure gradients (zero for this case). See SUBROUTINE NODPPR.

IARRAY 41 1 T sets IARRAY(41)=N2WAKE=1

N2WAKE alters the flow path in the execution module such that routines appropriate only to flows near solid boundaries are not called.

Define the fluid parameters.

RARRAY 27 6.7556 T sets the free-stream velocity (UINF) = 6.7556 ft/sec

RARRAY 10 1.935 T fluid density (RHOINF) = 1.935 lbf-sec-sec/(ft⁴)

RARRAY 38 21.1E-6 T fluid viscosity (XMUINF) in lbf-sec/ft-ft

Define the streamwise extent of computation.

RARRAY 43 1.0 T set reference length (REFL) = 1.0 ft

RARRAY 24 10.0 T set initial X-plane (TQ) = 10.0

RARRAY 35 20. T set solution distance (TD) = 20.

The final station will be TF=TQ+TD=30.

RARRAY 304 0.5 T set implicitness factor (CIMPETH)

RARRAY 196 0.5 T set implicitness factor (THETA)

RARRAY 7 .01 T set initial axial stepsize (HSINIT)

RARRAY 16 10. T set maximum allowable stepsize (HMAX)

HMAX is expected as a percentage to be applied to TD. In this case HMAX will be computed $HMAX = .10 * TD = 2$.

RARRAY 13 101. T set the intervals for printing or storing data (DELP)

This is also input as a percentage of TD and in this case will be computed to be greater than the sol'n distance. It will print at the final station TF=30. Further print controls will be implemented via the VPVSX and VX3ST command sequence used further along in the data set. Note that if a value less than 100., e.g., 25., had been used, output would result at .25*TD, .50*TD etc.

Define coefficients for conservation equations: see Appendix A.

Note difference in RARRAY command parameters to set entry = 0. See BDINT.

```
RARRAY 385 1. T UCMULT
RARRAY 348 1. T VCMULT
RARRAY 339 1. T PCFACT
RARRAY 346 1. T VLDMLT
RARRAY 340 1. T PPFACT
RARRAY 396 1. T OSG
RARRAY 394 1. T T2FIX
RARRAY 384 1. T ECMULT
RARRAY 353 -1. T GUMULT
RARRAY 282 0. 500 T U2STRS
RARRAY 281 0. 500 T U1STRS
RARRAY 399 0. 500 T TU1U2P
RARRAY 398 0. 500 T T2PFIK
RARRAY 283 0. 500 T EKSTRS
```

Define shear stress coefficients.

```
RARRAY 400 1. T C112
RARRAY 401 1. T C113
RARRAY 402 1. T C222
RARRAY 403 1. T C223
RARRAY 404 1. T C332
RARRAY 405 1. T C333
RARRAY 406 1. T C122
RARRAY 407 1. T C123
RARRAY 408 1. T C132
RARRAY 409 1. T C133
RARRAY 410 1. T C231
RARRAY 411 1. T C232
```

Set up finite element geometry.

NM=3 denotes that a 3D parabolic sol'n is to be obtained. 2D triangular finite elements will be constructed.

```
IARRAY 52 19 T set no. of rows of computational nodes (KROW) = 19
IARRAY 50 19 T set no. of columns of nodes (KCOL) = 19
IARRAY 55 400 T set NODE
```

NODE should be set slightly greater than KROW*KCOL to allow for sufficient storage allocation in dynamic dimensioning routine FEDIMN.

Removing the "C" from 1st col. in next 3 statements turns on debug prints.

```
CIARRAY 61 1, 86 1 T
CIARRAY 76 300 , 7 200, 6 3 T
CIARRAY 122 1 T
```

Removing the "C" in next 2 statements will print status of all non-zero IARRAY and RARRAY entries to this point and terminate execution.

```
CICOND T
CEXIT T
```

Call SUBROUTINE FEDIMN to allocate storage for variable length vectors.

```
FEDIMN T
```

Specify sequencing for solution of equation system. The following table identifies the dependent variables by integer numbers.

variable 1	streamwise velocity (U1)
2	vertical velocity (U2)
3	lateral velocity (U3)
4	stagnation enthalpy (H)
5	turbulence K. E. (TKE)
6	dissipation fct. (EPS)
7	perturbation pres. (PP)
8	complementary pres. (PC)
9	penalty fct. (PHI)

IPINT T fill the IPINT vector, read the next card image
1 5 6 2 3 7 8 9 0 0 , 1 -3 -3 -5 -5 5*0 , 10*I1 1 T

The first group of 10 integers above specifies the order in which the eqns. are solved at each iteration, i.e., U1,TKE,EPS,U2,U3 etc. Note that the stag. enthalpy is not computed. Also the complementary pres. is not computed in this case but is constant. The third group of 10 integers, 10*I1 1 (this string denotes the 10 integers beginning at 1 with each successive integer incremented by 1; 1,2,3 etc) re-labels the governing equations according to the sequencing specified in the first group, i.e., eqn(1) is identified with U1, eqn(2) with TKE etc. The second group of 10 integers specifies a staggered start for the computation. Eqn(1) (U1) is solved beginning at the first axial step. The 2nd and 3rd eqns. (TKE and EPS) are turned on at step 3. The 4th and 5th eqns. are turned on at step 5. There is no delay for the remaining eqns.

Construct the finite element network

LINK2 14 T call SUBROUTINE DSCRTZ to set up coordinates.
VX2SCL T compute node spacing for normal (X2) direction
0.0 9 -1.5 1. 9 -3. 1. T
VX1SCL T compute node spacing for lateral (X3) direction
0.0, 9 +1.5 1. 9 3. 1. T
NDECRD
1 19, 1 19 T

ELEM T set up element connections for above geometry.
DONE T leave DSCRTZ.

See Ref. 2 pp. 8-9 and Ref. 3 pp. 22-23 for explanation of the above command sequence.

This particular string will result in the construction of a 3X3 square domain with the origin of coordinates at the top left corner. The X3 (lateral) axis is spanned by 19 equi-spaced nodes (18 elements) from 0.0 to 3.0. The X2 axis is also spanned by 19 equi-spaced nodes from 0.0 to -3.0. There results 361 nodes and 648 elements.

Establish boundary conditions for dependent variables. These command data have a prescribed format. See Ref. 2 pp. 21-22, 26 and Ref. 3 pp. 18-19 for a general description of the boundary condition procedures. For any variable the default bc is zero gradient normal to boundary in question. It is only necessary to list the boundaries (for each variable in turn) where this should not be the case. For the flow situation considered, the left boundary is a plane of symmetry, the top boundary is a "rigid lid" free surface (equivalent to symmetry), and the bottom and right boundaries are in the free stream.

The next command is a list of boundary nodes. The positive direction is counter-clock-wise, e.g., -TOP creates a list of the node numbers for the top row beginning at the left most node (no. 1) and continuing to the last node in that row (no. 19).

```
IBORD
LEFT          2    BOTTOM          2    RIGHT          2    TOP          2
DONE
```

```
LINK1  2 T call NODELM to establish element-node connectivity table.
List variables and boundaries where value should be maintained at initial
level.
```

```
KBNO    2 T Fix var. 2 (U2) at U2=0 on top bndry.
-TOP      DONE
KBNO    3 T Fix var. 3 (U3) at U3=0 on left bndry.
-LEFT     DONE
KBNO    9 T Fix var. 9 (PHI) at PHI=0 at freestream bndrys.
BOTTOM    RIGHT          2    DONE
KBNO    7 T Fix var. 7 (PP) at PP=0 at fresstream bndrys.
BOTTOM    RIGHT          2    DONE
```

All variables on all boundaries not specifically referenced above will be subject to homogeneous Neumann bc's. These bc's allow entrainment from the freestream. Note that PHI=0 allows non-vanishing normal gradient and mass efflux or influx. U1, TKE, and EPS could as well be prescribed as fixed at the freestream boundaries (in this instance) with no significant difference in the computed results.

Set up formats for LU6 printed output.

The next two commands store data for the first and last pages of the LU6 printed output. Anything placed after the command and before "DONE" will appear on the appropriate page.

```
CONTITLE T Read title for last page.
          3D WAKE FLOW - - - DTNSRDC: DESTROYER IB ROT.
```

```
DONE
DESCRIPT 204 T Descriptive title for first page.
          3D WAKE FLOW - - - DTNSRDC: DESTROYER
```

```
DONE

The next set of commands designates which scalar data is to be printed
at the output stations and specifies titles and scale factors to be applied
to the data. See Ref. 2 pp. 12-15.
```

Fill the header title vector until "DONE".

```
DESCRIPT 332 T IOPAR vector
REFERENCE      ENGLISH-FT      ENGLISH-IN      M-K-S      C-G-S
LENGTH..... .FT..... .IN..... .M..... .CM.
VELOCITY..... .FT/SEC... .N.A..... .M/S..... .CM/SEC.
DENSITY..... .LBM/FT3... .N.A..... .KG/M3..... .G/CC...
TEMPERATURE... .RANKINE... .N.A..... .KELVIN.... .N.A
ENTHALPY..... .BTU/LBM... .N.A..... .KJ/KG..... .N.A
FROZ.SPEC.HEAT. .BTU/LBM-R.. .N.A..... .KJ/KG-K.... .N.A
VISCOSITY..... .LBM/FT-S... .N.A..... .NT-S/M2.... .POISE...
LOCAL PRESSURE. .PSF..... .PSI..... .UEDGE..... .CPU TIME..
..DPDX1.... .ENERGY... .INT. VAR... NWGEOM H'S ..H21...
..G23..... .X1/LREF.. .DX1/LREF.. .EPSILON.. .DX1M/LREF
```

REFL REY. NO.
DONE

Define location in RARRAY of scalars in accord with above titles.

IONUMB T
999, 5*200, 999, 200 4*43, 200 27 200 27 27, 200
10 200 10 10, 200 58 200 58 200, 200 97 200 97
200, 200 30 200 30 200,
200 38 200 2*38, 999, 36 2*36 63 300, 100 135 120 200 186 188,
11 12 14 85 47 T

Define location in RARRAY of scale factors applied to above scalar data.
Positive values are multipliers and negative values are divisors.

MPARA T
5*2, 2 2 162 164 163, 2 2 2 164 163, 2 2 2 170 174,
2 2 2 165 2, 2 -175 2 2 2, 2 2 2 176 2, 2 2 2 177 178,
-351 2 169 2 2, 3*2 108 2, 5*2, 5*2 T

Designate titles for profiles of output dependent variables until
"DONE". See Ref. 2, page 15.

DESCRIPT 203 T
U1 / UREF U2 / UREF U3 / UREF TKE/TKEREF EPS/DISSREF
U1'U1' U1'U2' U1'U3' U2'U2' U2'U3'
U3'U3' PP P PHI
DONE

Define variables to be output under above titles. Each entry is a composite
number which is decoded by the program. See Ref. 2, pp. 17-21 and 35.

IOSAVE T output vectors
1248 2248 3248 5248 6248 3271
4271 5271 6271 7271 8271 7248
8248 9248 T

Define scalar multipliers to be applied to each vector in turn. In this
case all vectors are multiplied by RARRAY(2) (which = 1.0) except the
12th entry (PP) which is multiplied by RARRAY(171).

IOMULT T scalar multipliers for IOSAVE vectors.
11*2 171 2*2 14*1 T

Specify axial locations for printed and stored output. The program in
SUBROUTINE DPSISQ designates a print at axial locations where the
external pressure is specified. Even though the freestream pressure
is constant in this flow, the print mechanism will be activated by
creating a pressure table.

The next command specifies a pressure table with 4 entries, pressure
equals ambient level at each location.

VPVSX
4*2116.8 T

The next command specifies the axial locations where the pressure is at
the above levels. A print on LU6 will always occur at the initial and
final stations.

VX3ST
10. 16. 22. 30. T

Call SUBROUTINE DIMEN to compute non-dimensional parameters.

LINK3 4 T DIMEN and INDEX

Call SUBROUTINE GEOMFL to generate the element matrices and vectors for the natural coordinate system. See Ref. 4, Chapt. 3.

LINK1 3 T GEOMFL

Call SUBROUTINE NODELM again to compute element thickness and area from data calculated in GEOMFL.

LINK1 2 T NODELM

Call SUBROUTINE TBLINP (NTYPE=7) to link with SUBROUTINE DTNSRDC and initialize U1,U2,U3,TKE,EPS on each node in finite-element domain. Data will be required from LU19 (file INPUT19) specified above and prepared in advance by the user.

LINK2 10 T Distribute 3D wake vel. vector and TKE, EPS

Call subroutines SETDIF and DFCFBL to initialize diffusion coefficients.

LINK5 6 T SETDIF-DFCFBL

Call SUBROUTINE RNLDST to initialize Reynolds stresses.

LINK5 4 T RNLDST

Initialization is complete. Transfer control to main drive loop.

Removing "C" from 1st col. in next 4 commands will cause a print of all non-zero IARRAY and RARRAY parameters, a node table print, and a listing of the initial fluid state. The program will terminate at "EXIT" before entering execution sequence.

CICOND

CLINK2 5 T NODES

CLINK2 6 T OUTPUT

CEXIT

Turn off any debug switches which may have been turned on. Note: they may be left on during execution at the expense of a great amount of output on LU6.

IARRAY 61 0, 86 0 T

IARRAY 76 0, 7 0, 6 0 T

IARRAY 122 0 T

QKNINT T Begin integration procedures.

EXIT T Terminate run at end of QKNINT call.

CASE END

APPENDIX C SAMPLE DATA FROM OUTPUT UNITS

1 IU8

Xt	time	1Ppmn	Ppmn	1Ppmx	Ppmx	1Pmn	Pmn	1Pmx	Fmx
0 1000E+02	0 1001E+02	1	0 0000E+00	1	0 0000E+00	1	0 0000E+00	1	0 0000E+00
0 1001E+02	0 1002E+02	79	0 1485E 03	65	0 2562E-05	1	0 0000E+00	1	0 0000E+00
0 1002E+02	0 1003E+02	79	0 1485E 03	65	0 2563E-05	1	0 0000E+00	1	0 0000E+00
0 1003E+02	0 1005E+02	79	0 1484E 03	65	0 2577E 05	1	0 0000E+00	1	0 0000E+00

etc

0 3000E+02	0 3000E+02	41	0 3929E 04	154	0 7480E 06	1	0 0000E+00	1	0 0000E+00
------------	------------	----	------------	-----	------------	---	------------	---	------------

2 IU9

Xt	time	dx	1dPpmn	dPpdxmn	1dPpmx	dPpdxmn	1dPpmx	dPpdxmn	dPpdx
0 1000E+02	0 1001E+02	0 1000E-01	361	0 0000E+00	361	0 0000E+00	0	0 0000E+00	0 0000E+00
0 1001E+02	0 1002E+02	0 1000E-01	79	0 3634E+00	65	0 6269E-02	0	0 0000E+00	0 0000E+00
0 1002E+02	0 1003E+02	0 1100E-01	42	0 1551E 03	3	0 8651E-04	0	0 0000E+00	0 0000E+00
0 1003E+02	0 1005E+02	0 1210E-01	43	0 5364E 03	3	0 4089E-03	0	0 0000E+00	0 0000E+00

etc

0 3000E+02	0 3000E+02	0 1658E+01	66	0 1220E 04	61	0 5167E 04	0	0 0000E+00	0 0000E+00
------------	------------	------------	----	------------	----	------------	---	------------	------------

3 IU16

nel	Xt	time	V21	V22	V11	V12	nvmax	Vthmx	theta	lemx	EPsmax
1	0 1000E+02	0 1001E+02	0 1347E+00	0 3658E 02	0 3880E-01	0 12250E 01	60	0 4860E 01	0 21157E+03	42	0 66264E 03
2	0 1001E+02	0 1002E+02	0 1347E+00	0 3658E 02	0 3880E-01	0 12114E 01	60	0 4860E 01	0 21157E+03	42	0 66264E 03
3	0 1002E+02	0 1003E+02	0 13467E+00	0 3658E 02	0 3880E-01	0 11966E 01	60	0 4860E 01	0 21157E+03	42	0 66264E 03
4	0 1003E+02	0 1004E+02	0 13463E+00	0 3651E 02	0 3880E-01	0 11805E-01	60	0 4860E 01	0 21157E+03	42	0 66170E 03
5	0 1004E+02	0 1006E+02	0 13458E+00	0 3642E 02	0 3880E-01	0 11630E 01	60	0 4860E 01	0 21157E+03	42	0 66066E 03

etc

39	0 3000E+02	0 3000E+02	0 3927E 01	0 1568E 02	0 5147E 02	0 10097E-03	100	0 13156E 01	0 60506E+02	42	0 83351E 04
----	------------	------------	------------	------------	------------	-------------	-----	-------------	-------------	----	-------------

4 IU17

nel	Xt	itmx	TKmx	lumx	U'mx	lumx	U1mn	U2mx	U2mn	U3mx	U3mn
1	0 1000E+02	42	0 5306E 02	81	0 1047E+00	5	0 1346E+00	81	0 4568E 01	79	0 3464E 01
2	0 1001E+02	42	0 5306E 02	81	0 1047E+00	5	0 1347E+00	81	0 4568E 01	79	0 3464E 01
3	0 1002E+02	42	0 5306E 02	81	0 1046E+00	5	0 1347E+00	81	0 4568E 01	79	0 3464E 01
4	0 1003E+02	42	0 5301E 02	81	0 1045E+00	5	0 1346E+00	81	0 4568E 01	79	0 3464E 01
5	0 1004E+02	42	0 5295E 02	81	0 1045E+00	5	0 1346E+00	81	0 4568E 01	79	0 3464E 01

etc

39	0 3000E+02	42	0 1783E 02	101	0 3814E 01	6	0 3928E 01	100	0 1145E 01	96	0 1004E 01
----	------------	----	------------	-----	------------	---	------------	-----	------------	----	------------

NOTES: All of the above files are written from subroutine STNPRNT. One record is written to each file during each axial step. Each unit for each file is 1 record and LU's 8 9 are written only if IU8 1.

APPENDIX C continued

S LU15

381 19 0.10000E+01 0.21100E 04 0.19350E+01 0.87556E+01 ← dimensionlizing scalars

1 0 00000E+00 0.00000E+00
2 0 00000E+00 0.16667E+00
3 0 00000E+00 0.33333E+00
4 0 00000E+00 0.50000E+00
5 0 00000E+00 0.66667E+00

(X2,X3) for each node

scalar data and nodal coordinates written
on the first pass only

etc

380 0 30000E+01 0 28333E+01
381 0 30000E+01 0 30000E+01

X STATION 0 10000E+02

0 12250E 01 0 00000E+00 0.00000E+00 0.24123E 02 0.20312E 03 0.24955E 05 0.23256E 02 0.19265E 02 0.13153E 05 0.40254E 03
0 27100E 01 0 00000E+00 0.38800E-01 0.28289E 02 0.25766E 03 0.27964E 04 0.61419E 03 0.19164E 02 0.34905E 04 0.91131E 03
0 74600E 01 0 00000E+00 0.35300E-01 0.36449E 02 0.37723E 03 0.32922E 02 0.10769E 03 0.85539E 03 0.19063E 02 0.12852E 04 0.20751E 02
0 11495E 00 0 00000E+00 0.29750E-01 0.34432E 02 0.34837E 03 0.33613E 02 0.62570E 04 0.38980E 03 0.15524E 02 0.11676E 03 0.20813E 02
0 13475E 00 0 00000E+00 0.21850E-01 0.32051E 02 0.31106E 03 0.30888E 02 0.61750E 04 0.44456E 05 0.14601E 02 0.13052E 03 0.20315E 02
0 10865E 00 0 00000E+00 0.54625E-02 0.36589E 02 0.37941E 03 0.31655E 02 0.25583E 03 0.59921E 03 0.18659E 02 0.92187E 05 0.22507E 02
0 48200E 01 0 00000E+00 0.13656E 02 0.22399E 02 0.18173E 03 0.19521E 02 0.19412E 03 0.45965E 03 0.12173E 02 0.60069E 04 0.12130E 02

etc

0 00000E+00 0 00000E+00 0.00000E+00 0.25000E 04 0.21429E 06 0.23445E 04 0.00000E+00 0.00000E+00 0.14115E 04 0.00000E+00 0.14115E 04
0 00000E+00 0 00000E+00 0.00000E+00 0.25000E 04 0.21429E 06 0.23445E 04 0.00000E+00 0.00000E+00 0.14115E 04 0.00000E+00 0.14115E 04
X STATION 0 16029E+02
0 19746E 01 0 00000E+00 0.00000E+00 0.21380E 02 0.13088E 03 0.20231E 02 0.28424E 04 0.11435E 03 0.15878E 02 0.31235E 05 0.79859E 03
0 26745E 01 0 00000E+00 0.14537E-01 0.22394E 02 0.14144E 03 0.20773E 02 0.46825E 04 0.20752E 03 0.15648E 02 0.22848E 04 0.94489E 03
0 49922E 01 0 00000E+00 0.22899E 01 0.25059E 02 0.16867E 03 0.22829E 02 0.42575E 04 0.37820E 03 0.14674E 02 0.90453E 04 0.13155E 02

etc

0 00000E+00 0 23558E 06 0 19331E 06 0.25000E 04 0.21429E 06 0.23445E 04 0.00000E+00 0.00000E+00 0.14115E 04 0.65742E 09 0.14114E 04
0 00000E+00 0 59282E 06 0.61608E 06 0.25000E 04 0.21429E 06 0.23445E 04 0.00000E+00 0.00000E+00 0.14116E 04 0.17759E 08 0.14117E 04
X STATION 0 22342E+02
0 29276E 01 0 00000E+00 0.00000E+00 0.17598E 02 0.82790E 04 0.16617E 02 0.31926E 04 0.81084E 04 0.11967E 02 0.22666E 05 0.77437E 03
0 32915E 01 0 00000E+00 0.72209E 02 0.18009E 02 0.85639E 04 0.16711E 02 0.34247E 04 0.97059E 04 0.11847E 02 0.28734E 04 0.85439E 03

etc

0 00000E+00 0 10141E 06 0.52051E 06 0.25000E 04 0.21429E 06 0.23443E 04 0.00000E+00 0.00000E+00 0.14115E 04 0.29649E 08 0.14116E 04
X STATION 0 30000E+02
0 28595E 01 0 00000E+00 0.00000E+00 0.14161E 02 0.51471E 04 0.13286E 02 0.45579E 04 0.22229E 04 0.87557E 03 0.17317E 05 0.71672E 03
0 30234E 01 0 00000E+00 0.26166E 02 0.14357E 02 0.52595E 04 0.13387E 02 0.51739E 04 0.34754E 04 0.87771E 03 0.16946E 04 0.74540E 03

NOTES. These data are written from subroutine PLTOUT at axial stations where KENT=1. On each call the XI
station is written followed by KROW,LCOL number of records. Each record consists of the values
of 11 fluid variables at that node

APPENDIX C continued

6 LU18

361 19 19 0.10000E+01 0.21100E-04 0.19350E+01 0.67558E+01 0.10000E+01 ← dimensionlizing scalars

1 0 00000E+00
2 0 16687E+00
3 0 33333E+00
4 0 50000E+00
5 0 66667E+00
6 0 83333E+00 ← X3 nodal coordinate for top row of nodes (X2=0) written
7 0 10000E+01 on 1st pass only
8 0 11667E+01
9 0 13333E+01
10 0 15000E+01
11 0 16667E+01
12 0 18333E+01
13 0 20000E+01
14 0 21667E+01
15 0 23333E+01
16 0 25000E+01
17 0 26667E+01
18 0 28333E+01
19 0 30000E+01

dx1

X STATION 0.10000E+02 0.10000E+01 0.10000E-01 1

0	12250E 01	0	00000E+00	0	00000E+00	0	24123E 02	0	20312E-03	0	24955E-02	0	23256E 05	0	54799E 03	0	19265E 02	0	13153E 05	0	40254E 03
0	27100E 01	0	00000E+00	0	38800E-01	0	28269E-02	0	25766E-03	0	27994E-02	0	46704E 04	0	61419E 03	0	19164E 02	0	34905E 04	0	91131E 03
0	74600E 01	0	00000E+00	0	35300E-01	0	36449E 02	0	37723E-03	0	32932E 02	0	10769E 03	0	65539E 03	0	19063E 02	0	12852E 04	0	20751E 02
0	11495E+00	0	00000E+00	0	29750E-01	0	34432E-02	0	34637E-03	0	33613E-02	0	62570E 04	0	38980E 03	0	15524E 02	0	11676E 03	0	20813E 02
0	13475E+00	0	00000E+00	0	21850E-01	0	32051E 02	0	31106E-03	0	30888E-02	0	61750E 04	0	44456E 05	0	14601E 02	0	13052E 03	0	20315E 02
0	10865E+00	0	00000E+00	0	54825E-02	0	36589E 02	0	37941E-03	0	31855E-02	0	25583E 03	0	59929E 03	0	18659E 02	0	92187E 05	0	22507E 02
0	48200E 01	0	00000E+00	0	13656E-02	0	22399E 02	0	18173E-03	0	19521E-02	0	19412E 03	0	45965E 03	0	12173E 02	0	60009E 04	0	12130E 02
0	28800E 01	0	00000E+00	0	00000E+00	0	12039E 02	0	71811E-04	0	11032E-02	0	77918E 04	0	13549E 04	0	67093E 03	0	16866E 05	0	67612E 03
0	49050E 01	0	00000E+00	0	00000E+00	0	12813E 02	0	79544E-04	0	11878E-02	0	90482E 04	0	74563E 04	0	72260E 03	0	84212E 05	0	71232E 03
0	37250E 01	0	00000E+00	0	00000E+00	0	11049E 02	0	82958E-04	0	10097E-02	0	95147E 04	0	13295E 03	0	61571E 03	0	23075E 04	0	60543E 03
0	13350E 01	0	00000E+00	0	00000E+00	0	62878E-03	0	26800E-04	0	56513E-03	0	51481E 04	0	10779E 03	0	34967E 03	0	17432E 04	0	33542E 03
0	00000E+00	0	00000E+00	0	00000E+00	0	30356E-03	0	90868E-05	0	28136E-03	0	84043E 05	0	22139E 04	0	17070E 03	0	23917E 05	0	16876E 03
0	00000E+00	0	00000E+00	0	00000E+00	0	21995E-03	0	55919E-05	0	20837E-03	0	00000E+00	0	00000E+00	0	12418E 03	0	00000E+00	0	12418E 03
0	00000E+00	0	00000E+00	0	00000E+00	0	14572E-03	0	30156E-05	0	13686E-03	0	00000E+00	0	00000E+00	0	82273E 04	0	00000E+00	0	82273E 04
0	00000E+00	0	00000E+00	0	00000E+00	0	25000E-04	0	21429E-06	0	23445E 04	0	00000E+00	0	00000E+00	0	14115E 04	0	00000E+00	0	14115E 04
0	00000E+00	0	00000E+00	0	00000E+00	0	25000E-04	0	21429E-06	0	23445E 04	0	00000E+00	0	00000E+00	0	14115E 04	0	00000E+00	0	14115E 04
0	00000E+00	0	00000E+00	0	00000E+00	0	25000E-04	0	21429E-06	0	23445E 04	0	00000E+00	0	00000E+00	0	14115E 04	0	00000E+00	0	14115E 04
0	00000E+00	0	00000E+00	0	00000E+00	0	25000E-04	0	21429E-06	0	23445E 04	0	00000E+00	0	00000E+00	0	14115E 04	0	00000E+00	0	14115E 04
0	06000E+00	0	00000E+00	0	00000E+00	0	25000E-04	0	21429E-06	0	23445E 04	0	00000E+00	0	00000E+00	0	14115E 04	0	00000E+00	0	14115E 04
X STATION 0.10010E+02 0.11000E-01 0.10000E-01 2																					
0	12114E 01	0	00000E+00	0	00000E+00	0	24123E 02	0	20312E-03	0	24986E 02	0	21546E 05	0	54423E 03	0	19265E 02	0	12102E 05	0	40365E 03
0	26966E 01	0	00000E+00	0	38800E-01	0	28269E 02	0	25766E 03	0	28001E 02	0	47094E 04	0	61210E 03	0	19164E 02	0	35058E 04	0	91198E 03
0	74505E 01	0	00000E+00	0	35300E-01	0	36449E-02	0	37723E-03	0	32921E 02	0	10833E 03	0	65570E 03	0	19063E 02	0	12579E 04	0	20750E 02
0	11488E+00	0	00000E+00	0	29750E 01	0	34432E 02	0	34637E 03	0	33613E 02	0	62480E 04	0	38991E 03	0	15524E 02	0	11675E 03	0	20813E 02

etc.

NOTES: These data are written from subroutine PITSUR at each axial step if ILU18=1. Each call writes the values of 11 fluid variables for each node in the plane of the mean free surface.

APPENDIX C continued

7 LU6

These data are written from subroutine REOUTP at axial locations specified under the VX3ST command.

STEP FOR INT. STARTUP.									
DEP. VAR.	VECTOR.	IPINT	2'NDP.	1	-3	3	5	5	10
11-	0	12-	0	14-	0	15-	0	17-	0
21-	0	13-	0	14-	0	15-	0	17-	0
LOCATION OF DEPENDENT VAR.									
1	1	2	4	3-	5	4-	0	5-	2
11-	0	12-	0	13-	0	14-	0	15-	0
21-	0	13-	0	14-	0	15-	0	17-	0
ADDRESSES OF DEP. VARIABLES.									
1	1	2	4	3-	5	4-	0	5-	2
11-	0	12-	0	13-	0	14-	0	15-	0
21-	0	13-	0	14-	0	15-	0	17-	0

5 VARIABLES BEING INTEGRATED.

1 5 6 2 3

8 VARIABLES IN SOLUTION.

1 5 6 2 3 7 8 9

ORDER OF CALLS AT END OF QKNUIN

LINKS(10)

PRINTOUT VARIABLES

1248	2248	3248	5248	6248	3271	4271	5271	6271	7271
8271	7248	8248	9248						

PRINTOUT VARIABLE MULTIPLIERS.

2	2	2	2	2	2	2	2	2	2
2	171	2	2	2	2	2	2	2	2

PRINTOUT VARIABLE FACTORS. (N)

N	EQ	1	STRAIGHT	PRINT.					
N	EQ	2	MULT	LAST	VEC	BY	THIS	ONE	STORE
N	EQ	3	ADD	TO	LAST	VEC.	STORE	IN	LAST
N	GE	4	RAISE	ENTRIES	TO	(N	2)	POWER	LOC.
N	LT	0	TAKE	NTH	ROOT	OF	ENTRIES	IF	GT
									0.0

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

Note the other output options available using the 2nd string of numbers under the LOHUIT command.

This page is written on the 1st pass only

7 LU6 continued

STATUS OF 409 IARRAY VALUES

1	ND	1	2	NTYPE	7	3	8	KPRINT	0	0	4	ISPLIT	0	0	5	NOHEGB	0
6	KODG	0	7	KOD5	0	0	13	KCOORD	0	0	9	NRSTRT	0	0	10	NUCHLT	0
11	NFTDOF	1	12	NCC	0	0	18	NPART	0	0	14	NELEM	648	0	15	NN	0
16	NNODE	361	17	NNS	0	0	23	NB	0	2	19	NOE	0	0	20	NFRNT	132
21	NRESEI	0	22	NC	0	0	28	IPASS	0	4	24	NTEL	3	0	25	KIND	4
26	KOUNT	1	27	NSKIP	400	0	33	NBUG	0	0	29	IRUN	0	0	30	NP	9
31	NEQ	8	32	IWLSEP	2	0	38	NOUTVC	0	8	34	LPRINT	100	0	35	NISICC	0
36	NFRSCC	0	37	NEWPR	8	0	43	NDB	0	0	39	NBASSH	0	0	40	NOUTS	1
41	NZWAKE	1	42	INPUT	5	0	48	NDOF	0	0	44	NSTORE	0	0	45	NEXF	0
46	NF	4	47	IG	2	0	53	NDB	0	0	49	NQEL	3	0	50	LCOL	18
51	IKOUT	0	52	KROW	19	0	58	NEQKNN	0	5	54	NHALF	0	0	55	NODE	400
56	MADDF	0	57	NEQXXX	6	0	63	NI	0	0	64	NSH	10	0	65	NHBSAV	0
61	KDUMP	0	62	NTITL	4	0	68	NI	0	0	69	NSEI	0	0	70	NDFSET	0
66	IFILIR	2	67	NS	0	0	73	ISTAD	0	0	74	IRRAY	0	0	75		0
71	ISTAIN	0	72	IVALUE	0	0	78	NGRPP	0	0	79	NQ	0	0	80	NPRDCT	1
76	KREDEG	3	77	NGRPHI	0	0	83	IFSLT	0	0	84	NVARD	0	0	85	NVARI	0
81	IWVAL	0	82	IVSTA	0	0	88	NLINE	0	60	89	NURUDT	0	0	90	NYI	4
86	KPHT	0	87	NRTAPE	0	0	93	NTR	0	2	94	NIZS	200	0	95	NVELTB	0
91	NZZ	4	92	IZSIZE	700000	0	98	IMAT	1	1	99	IBL	1	0	100	IREND	131616
96	IBASE	200	97	ITKE	1	0	103	HTH	0	0	104	NFLIP	1	0	105	IFTSPL	1
101	NJETS	0	102	KOUT	0	0	108	MCNADD	0	0	109	NTAPER	0	0	110	NFRSH	0
106	KK	0	107	NELE2	3	0	113	KFLVAR	0	0	114		0	0	115	IFORCE	0
111	IMAX	1	112	MWALLS	1	0	118	IBUG2	0	0	119	NGBUG	0	0	120	IFR	0
116		0	117	IBUG1	0	0	123	IGAS	0	0	124	NDERIV	2	0	125	NCALLS	2
121	NSPEC	10	122	IWRIT	0	0	128	ICNTND	0	0	129	INATAB	0	0	130	NGETH	1
126	ISPEED	0	127	IDIFRT	0	0	133	ITOP	0	79	134	NFSBDL	0	0	135	NFSBDL	0
131	NBORD	0	132	IPWRIT	0	0	138		0	0	139	INGAS	0	0	140	NTCNTS	0
136	KSAV	0	137		0	0	143	ICALL5	0	0	144	ITRAN	0	0	145	NICOND	1
141	KFSL	0	142	NOUTPR	59	0	148	NONC	0	0	149	IT	0	0	150	NODNO	0
146	NSFDBE	1	147	NPAST	0	0	153	NPUNCH	0	0	154	NSDFCE	0	0	155	NS2DFC	0
151	KWFLXS	0	152	INITKE	1	0	158	INITCN	0	0	159	IL	0	0	160	IH	0
156	ITWGLL	0	157	NPASS	0	0	163	NFSIST	0	0	164	NFSIND	0	0	165	NCALLI	0
161	NPVSX	4	162	IEPSET	99	0	168	NDRPRES	0	0	169	KCDC	0	0	170	NBC	0
166	N3DPNS	1	167	KNTPAS	0	0	173	NCOMHOS	0	0	174	NCOHTD	3	0	175	IFSL	0
171	NBCND	401	172	LOWD	2	0	178	NUSPOS	0	0	179	LOC	1	0	180	NSTRT	11
176	NKS	10	177	NUS2POS	0	0	183		0	0	184		0	0	185		0
181	NVELP	0	182	LOGS	0	0	188	KTFIN	0	0	189	NSTD	9	0	190	NMOUT	3
186	ITDA	0	187	ITDB	0	0	193	NM2	0	9	194	NDF	10	0	195	KXBND	0
191	NH	3	192	N2M	6	0	198	NPRDRT	0	4	199	NPRDRT	4	0	200	INTPSI	0
196	ITWALL	0	197	NTPRNT	0	0											

etc

401	ISPLTR	200	402	IEFLTR	200	403	IFIFLTR	1	404	IFINC	1	405	IFRSCAL	0
406	IFSRN	1	407	ILU8	1	408	ILU18	1	409		0			

The status of IARRAY is written on each call to REOUTP.

STATUS OF 414 BARRY VALUES.

1	FACT	1.0000E+00	2	ONE	1.0000E+00	3	ALC	4	THK	5	AJ	7	7828E+02
6	SOUND	0.0000E+00	7	HSINIT	1.0000E-02	8	RHINS1	9	PINF	10	RHOINF	15	1.9350E+00
11	XT	0.0000E+00	12	HI	0.0000E+00	13	DELP	14	EPS	15	H	20	1.0000E+02
16	HMAX	2.0000E+00	17	HMIN	0.0000E+00	18	U2INF	19	U3INF	20	PTIM	25	3.0200E+01
21	RE	6.1953E+05	22	TF	3.0000E+01	23	TINE	24	TO	25	CPOINF	30	0.0000E+00
26	TWOPI	6.2832E+00	27	UINF	6.7556E+00	28	RUNIV	29		30		35	7.7218E+00
31	G	1.0000E+00	32	RTCON1	1.5568E+03	33	XUINF	34	FEDGE	35	TD	40	2.0000E+01
36	PEDD1H	2.1168E+03	37	XLE	1.0000E+00	38	XUINF	39	SQ2	40	TRATIO	45	1.0000E+00
41	XMDXR	0.0000E+00	42	TKEDGE	6.1953E+05	43	REFL	44	RTEST	45	SSINIT	50	1.0000E+02
46	AMNUT	0.0000E+00	47	REFLRE	1.0000E+00	48	TINESV	49		50	RTCON4	55	0.0000E+00
51	RTCONS	0.0000E+00	52	XSCALE	0.0000E+00	53	YSCALE	54		55	GAMMAF	60	1.4000E+00
56	XHACHO	5.9696E-03	57	RTCON6	1.0000E+00	58	TOFINF	59	FACTHU	60		65	0.0000E+00
61	XMF	2.9400E+01	62	CONV	1.0000E+00	63	UEDGE	64	DIAGNL	65	CONRHO	70	1.0000E+00
66	STIDVR	3.8478E-07	67	PR	4.9200E+02	68	EP4ND	69	STLDEX	70	STLCON	75	2.1100E+05
71	STIDVR	3.8478E-07	72	STLDR	4.9200E+02	73	STLDR	74	STLDEX	75	STLCON	80	2.4297E+04
76	ZIT	0.0000E+00	77	CON1	9.9069E-03	78	CON2	79	FACTP	80	FACTH	85	0.0000E+00
81	UTAU	0.0000E+00	82	FIBAR	1.0000E+00	83	CONP	84	CONP	85	HMIN	90	0.0000E+00
86	AVD	2.5300E+01	87	U2REF	0.0000E+00	88	CONP	89	EPSINF	90	TKINF	95	4.5638E+01
91		0.0000E+00	92	PFSLL	1.0000E-01	93	PFSLL	94	RATQ2	95	EFTST	100	1.5000E+10
96	PSISTR	0.0000E+00	97	HINF	4.1157E+03	98	XPRIME	99	PPRCON	100	PPRIME	105	0.0000E+00
101		0.0000E+00	102		0.0000E+00	103	DEPLT	104	VELCST	105	ROCST	110	1.3370E+01
106	PTPL	3.0200E+01	107	VELD	0.0000E+00	108	ENMUL	109	XMA	110	XMH	115	2.6160E+00
111	TWELVE	1.2000E+01	112		0.0000E+00	113	XMSDF	114	ROUALC	115	ARSCAL	120	0.0000E+00
116	RR	1.4354E+01	117	RTOHM1	1.3083E+01	118	XMSDF	119	ROUALC	120	VARB	125	0.0000E+00
121	HDOT	0.0000E+00	122	HRSOT	0.0000E+00	123	QR	124	CON	125	XIAM	130	1.1000E+01
126		0.0000E+00	127		0.0000E+00	128	PCNT	129	SC	130	ENERGY	135	0.0000E+00
131	SPLIT	2.8350E+02	132	HRCON	2.9126E+02	133	RUESQ	134	Q3MAX	135	ENERGY	140	0.0000E+00
136	OSMAX	3.0000E+00	137	AVDP	0.0000E+00	138	RUESQ	139	H31	140	G32	145	0.0000E+04
141	G33	0.0000E+00	142	G1	5.2000E+02	143	CVH	144	CVU	145	E1E2SW	150	3.0000E+04
146	TOA	5.3000E+02	147	TOH	5.2000E+02	148	CVH	149	CVU	150	CVT	155	1.0000E+00
151	CVP	4.7250E-04	152	CVRHO	1.6020E+01	153	CVCP	154	XHACHS	155	TSINF	160	5.3300E+02
156	AINF	1.1317E+03	157	RHOIN	1.3072E+01	158	CFA	159	CPH	160	CPINF	165	7.7218E+00
161	PRFACT	0.0000E+00	162	FTOIN	1.2000E+01	163	FTTOCH	164	FTTOCH	165	DRTOCK	170	5.5556E+01
166	PSFTOA	4.7250E-04	167	PSFTOT	3.5910E+01	168	PSFTON	169	PSFTOI	170	PDFTOK	175	1.6020E+01
171	EULER	2.4470E+01	172	XHFACT	6.0138E+03	173	VLBTOP	174	PDFTOC	175	EBTOKJ	180	2.3244E+00
176	CBTOKJ	4.1840E+00	177	VLBTON	1.4880E+00	178	VLBTOP	179	RADCON	180	FMSKGS	185	4.5455E+01
181	CIKORE	1.0000E+00	182	CIKORE	1.4400E+00	183	C2DORF	184	C2KORE	185	PRDIS	190	1.3000E+00
186	H21	1.0000E+00	187	G22	0.0000E+00	188	G23	189	F1	190	SLOPE	195	0.0000E+00
191	ADUCT	1.0000E+00	192	G22	0.0000E+00	193	XNWGEO	194	XTTT	195	YTT	200	0.0000E+00
196	THETAP	5.0000E+01	197		0.0000E+00	198	YPLUS	199	RNULOC	200			0.0000E+00
etc													
401	c113	1.0000E+00	402	c222	1.0000E+00	403	c223	404	c332	405	c333		1.0000E+00
406	c122	1.0000E+00	407	c123	1.0000E+00	408	c132	409	c133	410	c231		1.0000E+00
411	c232	1.0000E+00	412		1.0000E+00	413		414					0.0000E+00

The status of BARRY is written on each call to REOUTP.

7. LU6 continued

NODE AND COORDINATE MAP OF DISCRETIZATION

X2COR
SCALE
FACTOR

X2COR
NODE NUMBERS

X1COR
SCALE
FACTOR

X2 scale and coordinate

Node nos.

E	1	001	002	003	004	005	006	007	008	009	010	011	012	013	014
0	166667	020	021	022	023	024	025	026	027	028	029	030	031	032	033
0333333	039	040	041	042	043	044	045	046	047	048	049	050	051	052	053
0500000	058	059	060	061	062	063	064	065	066	067	068	069	070	071	072
0666667	077	078	079	080	081	082	083	084	085	086	087	088	089	090	091
0833333	096	097	098	099	100	101	102	103	104	105	106	107	108	109	110
1000000	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129
1166667	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148
1333333	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167
1500000	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186
1666667	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205
1833333	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224
2000000	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243
2166666	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262
2333333	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281
2500000	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
2666667	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319
2833334	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338
3000000	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357

E 1 0 0166667 0333333 0500000 0666667 0833333 1000000 1166667 1333333 1500000 1666667 1833333 2000000 2166666

X3 scale and coordinate

E	1	015	016	017	018	019
0	166667	034	035	036	037	038
0333333	053	054	055	056	057	058
0500000	072	073	074	075	076	077
0666667	091	092	093	094	095	096
0833333	110	111	112	113	114	115
1000000	129	130	131	132	133	134
1166667	148	149	150	151	152	153

etc

2833334	338	339	340	341	342
3000000	357	358	359	360	361

E 1 2333333 2500000 2666667 2833334 3000000 X3 coordinate

X3 scale

NOTE: Real data are written in "E" format with the leading decimal and and exponent designation omitted. The correct scaling is printed by the associated column or row.

3-DIMENSIONAL PARABOLIC NAVIER STOKES OPTION

33D WAKE FLOW - - DTNSRDC: DESTROYER

REFERENCE	ENGLISH FT	ENGLISH IN	M-K S	C-G S
X X X X X	X X X X X	X X X X X	X X X X X	X X X X X
LENGTH	FT.	IN.	M.	CM.
VELOCITY	1000000E+01	1200000E+02	3048000E+00	3048000E+02
DENSITY	FT/SEC.	N.A.	M/S	CM/SEC.
TEMPERATURE	6785600E+01	N.A.	2059107E+01	2059107E+03
ENTHALPY	LBM/FT3	N.A.	KG/M3	G/CC.
FROZ. SPEC HEAT	1935000E+01	N.A.	309870E+02	309870E+01
VISCOSITY	RANKINE	N.A.	KELVIN	N.A.
LOCAL PRESSURE	5330000E+03	N.A.	KJ/KG	N.A.
DPDX1	BTU/LBM	N.A.	4115698E+04	N.A.
G23	BTU/LBM.R	N.A.	KJ/KG-K	N.A.
REFL. REV. NO	7721760E+01	N.A.	3230784E+02	POISE
X X X X X	X X X X X	X X X X X	X X X X X	X X X X X
LENGTH	FT.	IN.	M.	CM.
VELOCITY	1000000E+01	1200000E+02	3048000E+00	3048000E+02
DENSITY	FT/SEC.	N.A.	M/S	CM/SEC.
TEMPERATURE	6785600E+01	N.A.	2059107E+01	2059107E+03
ENTHALPY	LBM/FT3	N.A.	KG/M3	G/CC.
FROZ. SPEC HEAT	1935000E+01	N.A.	309870E+02	309870E+01
VISCOSITY	RANKINE	N.A.	KELVIN	N.A.
LOCAL PRESSURE	5330000E+03	N.A.	KJ/KG	N.A.
DPDX1	BTU/LBM	N.A.	4115698E+04	N.A.
G23	BTU/LBM.R	N.A.	KJ/KG-K	N.A.
REFL. REV. NO	7721760E+01	N.A.	3230784E+02	POISE
X X X X X	X X X X X	X X X X X	X X X X X	X X X X X
LENGTH	FT.	IN.	M.	CM.
VELOCITY	1000000E+01	1200000E+02	3048000E+00	3048000E+02
DENSITY	FT/SEC.	N.A.	M/S	CM/SEC.
TEMPERATURE	6785600E+01	N.A.	2059107E+01	2059107E+03
ENTHALPY	LBM/FT3	N.A.	KG/M3	G/CC.
FROZ. SPEC HEAT	1935000E+01	N.A.	309870E+02	309870E+01
VISCOSITY	RANKINE	N.A.	KELVIN	N.A.
LOCAL PRESSURE	5330000E+03	N.A.	KJ/KG	N.A.
DPDX1	BTU/LBM	N.A.	4115698E+04	N.A.
G23	BTU/LBM.R	N.A.	KJ/KG-K	N.A.
REFL. REV. NO	7721760E+01	N.A.	3230784E+02	POISE
X X X X X	X X X X X	X X X X X	X X X X X	X X X X X
LENGTH	FT.	IN.	M.	CM.
VELOCITY	1000000E+01	1200000E+02	3048000E+00	3048000E+02
DENSITY	FT/SEC.	N.A.	M/S	CM/SEC.
TEMPERATURE	6785600E+01	N.A.	2059107E+01	2059107E+03
ENTHALPY	LBM/FT3	N.A.	KG/M3	G/CC.
FROZ. SPEC HEAT	1935000E+01	N.A.	309870E+02	309870E+01
VISCOSITY	RANKINE	N.A.	KELVIN	N.A.
LOCAL PRESSURE	5330000E+03	N.A.	KJ/KG	N.A.
DPDX1	BTU/LBM	N.A.	4115698E+04	N.A.
G23	BTU/LBM.R	N.A.	KJ/KG-K	N.A.
REFL. REV. NO	7721760E+01	N.A.	3230784E+02	POISE
X X X X X	X X X X X	X X X X X	X X X X X	X X X X X
LENGTH	FT.	IN.	M.	CM.
VELOCITY	1000000E+01	1200000E+02	3048000E+00	3048000E+02
DENSITY	FT/SEC.	N.A.	M/S	CM/SEC.
TEMPERATURE	6785600E+01	N.A.	2059107E+01	2059107E+03
ENTHALPY	LBM/FT3	N.A.	KG/M3	G/CC.
FROZ. SPEC HEAT	1935000E+01	N.A.	309870E+02	309870E+01
VISCOSITY	RANKINE	N.A.	KELVIN	N.A.
LOCAL PRESSURE	5330000E+03	N.A.	KJ/KG	N.A.
DPDX1	BTU/LBM	N.A.	4115698E+04	N.A.
G23	BTU/LBM.R	N.A.	KJ/KG-K	N.A.
REFL. REV. NO	7721760E+01	N.A.	3230784E+02	POISE
X X X X X	X X X X X	X X X X X	X X X X X	X X X X X
LENGTH	FT.	IN.	M.	CM.
VELOCITY	1000000E+01	1200000E+02	3048000E+00	3048000E+02
DENSITY	FT/SEC.	N.A.	M/S	CM/SEC.
TEMPERATURE	6785600E+01	N.A.	2059107E+01	2059107E+03
ENTHALPY	LBM/FT3	N.A.	KG/M3	G/CC.
FROZ. SPEC HEAT	1935000E+01	N.A.	309870E+02	309870E+01
VISCOSITY	RANKINE	N.A.	KELVIN	N.A.
LOCAL PRESSURE	5330000E+03	N.A.	KJ/KG	N.A.
DPDX1	BTU/LBM	N.A.	4115698E+04	N.A.
G23	BTU/LBM.R	N.A.	KJ/KG-K	N.A.
REFL. REV. NO	7721760E+01	N.A.	3230784E+02	POISE
X X X X X	X X X X X	X X X X X	X X X X X	X X X X X
LENGTH	FT.	IN.	M.	CM.
VELOCITY	1000000E+01	1200000E+02	3048000E+00	3048000E+02
DENSITY	FT/SEC.	N.A.	M/S	CM/SEC.
TEMPERATURE	6785600E+01	N.A.	2059107E+01	

Each call to **REOUTP** results in print of above header data and full planar output for each vector listed under **IOSAVE** command. See the next page for an example of a typical vector output for this geometry

7. LU6 continued

E 1	UI / UREF		E 1		UI scale		UI at each node	
	UI	UREF	E 1	UI	UI scale	UI at each node	UI	UI at each node
0	101250	0972900	0925400	0885050	0867350	0891350	0951800	0971200
-0166667	1011930	0966940	0917020	0881050	0873610	0912430	0968320	0978660
0333333	1004240	0944910	0914600	0931085	0960770	0985280	0999365	0998180
0500000	1014590	0963205	0987725	1060200	1077030	1044810	1011430	1005880
0666667	1038790	1014640	1038690	1090100	1104710	1078400	1023500	1005510
0833333	1027840	1035340	1054100	1071590	1091130	1070140	1022160	1009500
1000000	1012810	1019220	1029070	1041290	1058570	1038210	1010370	1006030
-1166667	1008430	1009060	1013040	1018250	1022110	1015510	1009250	1007180
-1333333	1002110	1002260	1003280	1004560	1005530	1003680	1002310	1001800
-1500000	1000530	1000570	1000810	1001140	1001380	1000970	1000580	1000450
-1666667	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
1833333	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
-2000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
2166668	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
2333333	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
2500000	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
2666667	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
2833334	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
3000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
E 1	0	0166667	0333333	0500000	0666667	0833333	1000000	1166667
							1333333	1500000
							1666667	1833333
							2000000	2166666
0	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
0166667	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
0333333	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
0500000	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
0666667	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
0833333	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
1166667	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
1333333	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
1500000	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
1666667	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
1833333	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
2000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
2166668	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
2333333	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
2500000	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
2666667	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
2833334	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
3000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
E 1	2333333	2500000	2666667	2833334	3000000			

END

DATE

FILMED

DTIC

JULY 88